

Universidad de Alcalá

Escuela Politécnica Superior

Grado en Ingeniería en Tecnologías de la Telecomunicación

Trabajo de Fin de Grado

Aplicación de trazabilidad de productos en líneas de fabricación
de cosmética

Autora: María Luengo Rodas

Tutora: Marta Marrón Romera

Cotutora: Cristina de Evan de Marcos

Curso 2020/2021

UNIVERSIDAD DE ALCALÁ

ESCUELA POLITÉCNICA SUPERIOR

Grado en Ingeniería en Tecnologías de la Telecomunicación

Trabajo de Fin de Grado

**Aplicación de trazabilidad de productos en líneas de fabricación
de cosmética**

Autora: María Luengo Rodas

Tutora: Marta Marrón Romera

Cotutora: Cristina de Evan de Marcos

Tribunal:

Presidente: D. Oscar Esteban Martínez

Vocal 1: Dña. Sonia Martín López

Vocal 2: Dña. Marta Marrón Romera

Calificación:

Fecha: 30 de septiembre de 2021

A mi hermana, Sara

Agradecimientos

A pesar de las dificultades que hemos tenido a la hora de realizar este TFG, quiero dar las gracias a mi tutora Marta, y a mi cotutora Cristina por confiar en mí y darme la oportunidad de hacer un proyecto que se salía un poco de mi zona de confort. La ingeniería industrial, la automatización y la robótica no estaban en mi mente antes de realizar este trabajo, pero, a medida que fui conociendo más sobre ese mundo, me he vuelto una completa apasionada, y, ahora, después de estos largos y duros años de carrera, estoy más segura que nunca de mis nuevos objetivos profesionales, he encontrado mi pasión.

Dar las gracias, también, a mis compañeros de universidad que con el tiempo se convirtieron en amigos, os guardaré siempre un lugar en mi vida a pesar de que nuestros caminos hoy se separen. En especial, agradecer a mi amiga Andrea por todo lo que ha hecho por mí estos años, por ser mi ejemplo a seguir, por ayudarme, apoyarme y creer en mí. A Irene por su risa contagiosa y por no dejarme sola ni en los buenos ni en los malos momentos, y, por último, a Marta, gracias por ponernos siempre un pie en la tierra. Gracias a las tres por darme una amistad sana y bonita, fuera de competencias, envidias y egoísmos, sin duda, lo mejor que me ha dado la carrera ha sido conocerlos, sin vosotras nada hubiera sido lo mismo.

Nada de esto estaría escribiendo ahora si no hubiera tenido el apoyo de mis padres durante estos 23 años. El mayor agradecimiento va para ellos, por hacerme ver que en los estudios siempre he sido suficiente, que no pasaba nada por sacar un 6, un 7 o un 10 si me había esforzado al 100 % para conseguirlo. Por mostrarme siempre que no estudio para ellos si no por mí y para mí, porque mi futuro soy yo, aunque espero que ellos formen parte de él muchísimos años más. Os quiero.

Resumen

Este Trabajo de Fin de Grado (TFG) se centra en los conceptos de *Automatización* y *Trazabilidad* en la industria, más concretamente en la industria de cosmética, y la importancia que tiene implementar sus soluciones en las líneas de producción. Para entender de forma más realista el concepto de *Automatización* se ha desarrollado, con la ayuda de Sysmac Studio (una plataforma para la automatización de líneas de producción de la empresa Omron) un robot SCARA en 3D que lleva a cabo el movimiento de coger y dejar una pieza, similar al que realizaría un robot en una línea de producción real, por ejemplo, cuando se necesita retirar una pieza defectuosa de la cinta transportadora. Utilizando el mismo software y el sensor de visión FQ-M de Omron se han utilizado dos métodos de detección de piezas, para entender mejor el concepto de *Trazabilidad*.

Palabras clave: Automatización; Trazabilidad; Robot SCARA; Sysmac Studio; Sensor de Visión.

Abstract

This Final Degree Project (TFG) focuses on the concepts of *Automation* and *Traceability* in the industry, more specifically in the cosmetics industry, and the importance of implementing its solutions in production lines. To understand the concept of *Automation* more realistically, a 3D SCARA robot has been developed with the help of Sysmac Studio (a platform for the automation of production lines of the Omron company) that carries out the movement of picking up and leaving a part, similar to what a robot would perform on a real production line, for example, when a defective part needs to be removed from the conveyor belt. Using the same software and Omron's FQ-M vision sensor, two part detection methods have been used to understand better the concept of *Traceability*.

Keywords: Automation; Traceability; SCARA robot; Sysmac Studio; Vision Sensor.

Índice

Resumen	ix
Abstract	xi
Índice	xiii
Lista de figuras	xv
Lista de tablas	xix
Glosario	xxi
1 Introducción	1
1.1 Motivación	1
1.2 Objetivos	4
1.3 Estructura	5
2 Base Teórica	7
2.1 Automatización	8
2.2 Trazabilidad	10
2.2.1 Visión artificial	14
2.2.2 Código de barras	16
2.2.3 RFID	19
2.3 Aplicación Pick&Place de Omron	21
2.4 Sysmac Studio	27
3 Desarrollo	35
3.1 Aplicación de Trazabilidad	35

3.1.1	Conexión de software y máquina	35
3.1.2	Calibración	37
3.1.3	Métodos de detección	41
3.1.4	Transmisión de datos de salida	44
3.2	Aplicación de Pick&Place	47
3.2.1	Creación del proyecto y registro de ejes	49
3.2.2	Creación de Datos de Forma 3D	52
3.2.3	Creación del programa para el movimiento del Robot	57
4	Resultados obtenidos	59
4.1	Resultados de Trazabilidad	59
4.1.1	Búsqueda de forma	59
4.1.2	Etiquetado	63
4.2	Resultados de Pick&Place	67
4.2.1	Montaje completo Robot SCARA 3D	67
4.2.2	Programa para el movimiento del ROBOT	68
5	Conclusiones y trabajos futuros	75
5.1	Conclusiones	75
5.2	Trabajos futuros	75
6	Presupuesto	77
7	Pliego de condiciones	79
	Apéndices	81
	Anexos	81
A	BLOQUE DE FUNCIÓN: MC_SetKinTransform	81
B	BLOQUE DE FUNCIÓN: MC_MoveTimeAbsolute	81
	Bibliografía	87

Lista de figuras

1.1	Número de empresas activas en España de 2007 a 2020 (Fuente: [1])	1
1.2	Gasto en tecnologías y servicios de transformación digital en todo el mundo de 2017 a 2024. (Fuente: [2])	2
2.1	Partes de un proceso de automatización.	9
2.2	Trazabilidad hacia atrás en dirección al actor implicado, “aguas arriba”.	11
2.3	Trazabilidad interna o de proceso.	12
2.4	Trazabilidad hacia adelante a partir del actor implicado, “aguas abajo”.	13
2.5	Ejemplo de cámara de visión artificial, en aplicaciones de trazabilidad industrial. (Fuente: [3])	15
2.6	Ejemplo de códigos de barras 1D y 2D. (Fuente: [4])	16
2.7	Tipos de códigos de barras.	17
2.8	Ejemplo de etiqueta RFID. (Fuente: Century Europe)	19
2.9	Elementos de un sistema RFID. (Fuente: [5])	20
2.10	Robot Demo aplicación Pick&Place de la empresa Omron.	22
2.11	Pantalla NS, modelo NS15/NS12/NS10/NS8 (Fuente: [6]).	22
2.12	Controlador de máquina NJ501 - 4500. (Fuente:[6])	23
2.13	Sensor de visión FQ-M. (Fuente:[6])	23
2.14	Estructura cinemática de un robot, a) serie, b) paralelo (Fuente [7])	25
2.15	Robot X-Delta 3+1. (Fuente:[6])	26
2.16	Ejes de rotación y desplazamiento del robot. (Fuente:[8])	27
2.17	Robot SCARA Serie X. (Fuente:[6])	27
2.18	Solución completa: Plataforma de automatización integrada. (Fuente:[6])	28
2.19	Modelo Software IEC 61131-3.	29
2.20	Esquema de un programa asignado a una tarea.	29

2.21 Estructuras general de una función (FUN) y de un bloque de función (FB)	30
2.22 Bloque de función «TON» en lenguaje <i>Ladder</i> y <i>ST</i>	31
2.23 Función «MAX» en lenguaje <i>Ladder</i> y <i>ST</i>	31
2.24 Tipo de variables definidas por el usuario.	32
2.25 Ventana principal Software Sysmac Studio.	32
2.26 Bastidor CPU.	33
2.27 Módulo de entradas y salidas añadido al controlador.	33
2.28 Red EtherCAT con todos los dispositivos de la máquina real.	34
2.29 Insertar una FUN o un FB en el programa.	34
3.1 Conexión Ethernet software-máquina.	36
3.2 Red EtherCAT con conexión al sensor FQ-M de la máquina real.	36
3.3 Pantalla de trabajo FQ-M.	37
3.4 Flujo de trabajo para realizar la calibración <i>Muestro por lote</i>	38
3.5 Plantilla para la calibración.	38
3.6 Ruta para acceder al asistente de calibración.	39
3.7 Configuración de tipo de calibración.	39
3.8 Configuración número de puntos de muestreo.	40
3.9 Identificación del objeto de calibración (plantilla).	40
3.10 Muestreo aguas arriba.	41
3.11 Anotación de las coordenadas del robot	41
3.12 Muestreo aguas abajo.	41
3.13 Resumen elementos de inspección.	42
3.14 Método de detección: Búsqueda de forma.	42
3.15 Método de detección: Búsqueda.	43
3.16 Método de detección: Posición por bordes.	43
3.17 Método de detección: Etiquetado.	44
3.18 Comunicación comando/respuesta: Sensor de visión - Controlador.	45
3.19 Salida de datos: Sensor de visión - Controlador.	45
3.20 Configuración de la comunicación EtherCAT.	46
3.21 Ventana de configuración de la red de dirección del nodo.	46
3.22 Edición de ajustes Mapa PDO.	47

3.23	Mapa entradas y salidas del Sensor de Visión.	47
3.24	Partes ROBOT SCARA.	48
3.25	Coordenadas ROBOT SCARA.	48
3.26	Cinemática ROBOT SCARA.	49
3.27	Pasos para el desarrollo del ROBOT SCARA.	49
3.28	Añadir un eje nuevo al sistema.	50
3.29	Configuración básica del eje.	50
3.30	Configuración de Conversión de unidades.	51
3.31	Configuración de operación.	51
3.32	Creación grupo de ejes.	51
3.33	Configuración básica de grupo de ejes.	52
3.34	Creación de un Robot SCARA utilizando Sysmac Studio 3D.	52
3.35	Añadir un componente mecánico.	53
3.36	Relación entre componente mecánico y eje del robot.	53
3.37	Configuración de la BASE del robot SCARA.	53
3.38	Configuración de parámetros del BRAZO del robot SCARA.	54
3.39	Configuración de eje de rotación del BRAZO del robot SCARA.	54
3.40	Configuración común del BRAZO del robot SCARA.	54
3.41	Configuración de parámetros del ANTEBRAZO del robot SCARA.	55
3.42	Configuración común del ANTEBRAZO del robot SCARA.	55
3.43	Configuración de parámetros del EJE Z del robot SCARA.	55
3.44	Configuración de dirección lineal del EJE Z del robot SCARA.	56
3.45	Configuración común del EJE Z del robot SCARA.	56
3.46	Configuración de parámetros del EJE R del robot SCARA.	56
3.47	Configuración común del EJE Z del robot SCARA.	57
3.48	Bloque de función: MC_Power en Ladder.	57
3.49	Bloque de función: MC_GroupEnable en Ladder.	57
3.50	Bloque de función: MC_SetKinTransform en Ladder.	58
3.51	Bloque de función: MC_MoveTimeAbsolute en Ladder.	58
4.1	Pasos para la configuración del método de detección	60
4.2	Registro del elemento de inspección.	60

4.3	Reproducción de la escena del sensor de visión en el simulador. Modo RUN.	60
4.4	Registrar la región del modelo.	61
4.5	Determinar la región de medición.	61
4.6	Determinar el punto de detección.	62
4.7	Nivel de correlación del modelo con las piezas de 76.	62
4.8	Nivel de correlación del modelo con las piezas de 98.	63
4.9	Detección de piezas mediante el método: BÚSQUEDA DE FORMA.	63
4.10	Registro del elemento de inspección: Etiquetado	64
4.11	Selección de píxeles de una pieza roja.	64
4.12	Selección de píxeles de una pieza roja.	65
4.13	Determinar la región de medición.	65
4.14	Condiciones de medida: rellenado de orificios.	66
4.15	Detección de una pieza con do etiquetas.	66
4.16	Detección de piezas mediante el método: ETIQUETADO	66
4.17	Montaje del robot SCARA 3D con Sysmac Studio 3D.	67
4.18	Pasos en la simulación para el movimiento del robot.	68
4.19	Secciones creadas en el programa principal para el movimiento del robot.	68
4.20	Sección Power_ON → <i>Paso 1</i>	69
4.21	Inicialización de los parámetros cinemáticos del robot SCARA en texto estructurado. . .	70
4.22	<i>Workspace</i> de un robot SCARA	71
4.23	Inicialización de los parámetros del <i>Workspace</i> del robot SCARA en texto estructurado. .	71
4.24	Sección Establecer_cinemática → <i>Paso 2 y 3</i>	71
4.25	Inicialización de los parámetros para el movimiento del robot SCARA.	72
4.26	Sección Secuencia → <i>Paso 4</i>	72
4.27	Movimiento de Pick&Place Robot SCARA RPR + R en 3D	73
A.1	Manual de usuario NJ Robotics CPU Unit → Details of Robot Instructions (páginas 126-129)	82
A.2	Manual de usuario NJ Robotics CPU Unit → Details of Robot Instructions (páginas 126-129)	83
B.1	Manual de usuario NJ Robotics CPU Unit → Details of Robot Instructions (páginas 151-153)	84
B.2	Manual de usuario NJ Robotics CPU Unit → Details of Robot Instructions (páginas 151-153)	85

Lista de tablas

6.1	Coste del material físico.	77
6.2	Coste del software.	77
6.3	Coste total del material.	77
6.4	Coste de mano de obra.	78
6.5	Coste total del proyecto	78
6.6	Presupuesto de ejecución.	78
6.7	Presupuesto final del proyecto.	78

Glosario

Lista de acrónimos

TFG	Trabajo de Fin de Grado
TC	Tiempo de Ciclo (minuto/pieza)
Cap	Capacidad de producción (piezas/hora)
WS	Tiempo o Work Estándar (minutos de hombre/pieza)
P	Productividad (hora de hombre/pieza)
RFDI	Radio Frequency Identification
PLC	Controlador Lógico Programable
HMI	Interfaz Hombre-Máquina
IoT	Internet de las cosas
TFT	Thin Film Transistor
SCARA	Selective Compliance Assembly Robot
FUN	Función
FB	Bloque de función
POU	Program Organization Unit

Capítulo 1

Introducción

1.1 Motivación

Según la Empresa Statista en 2020 se registró en España el valor más alto de sociedades activas de la segunda década del siglo XXI con un total de 3.404.428 millones [1], figura 1.1. Con esa cantidad de empresas es difícil pensar que todas ellas presentan algo en común, pero, en efecto, el principal objetivo que comparten es el de buscar la mayor eficiencia y productividad posibles.

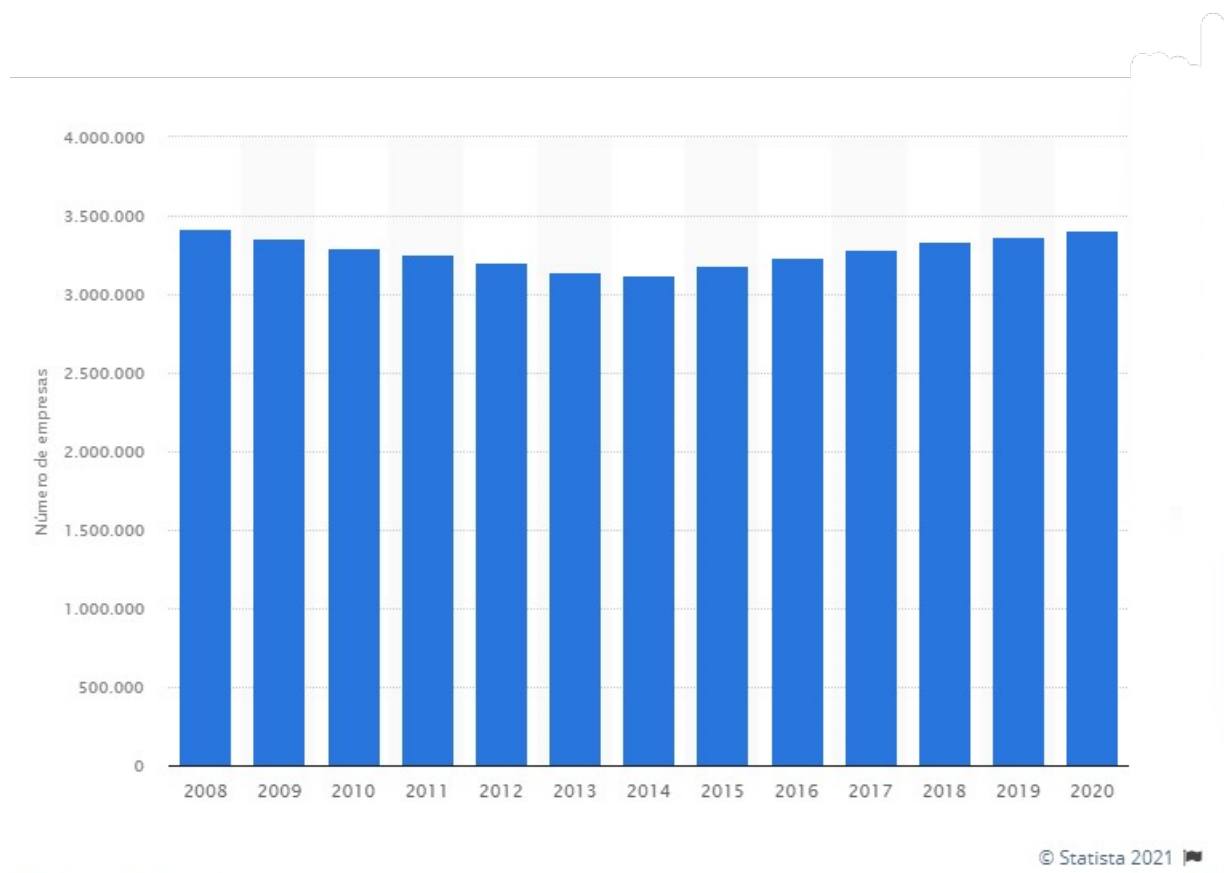


Figura 1.1: Número de empresas activas en España de 2007 a 2020 (Fuente: [1])

Entre las herramientas para conseguir dichos objetivos el desarrollo tecnológico juega el papel más importante. Las empresas invierten en innovaciones tecnológicas que a la larga traen consigo un crecimiento en la productividad y, por ende, en su crecimiento económico [9]. Según Statista [2] se estima que para el año 2023 las empresas gasten aproximadamente entre 2-3 billones de euros en tecnología y servicios que permitan la transformación digital, figura 1.2. Expertos advierten a las empresas sobre el peligro de no adaptarse a este nuevo mundo tecnológico, que podría conllevar un fracaso inmediato de las mismas [10].

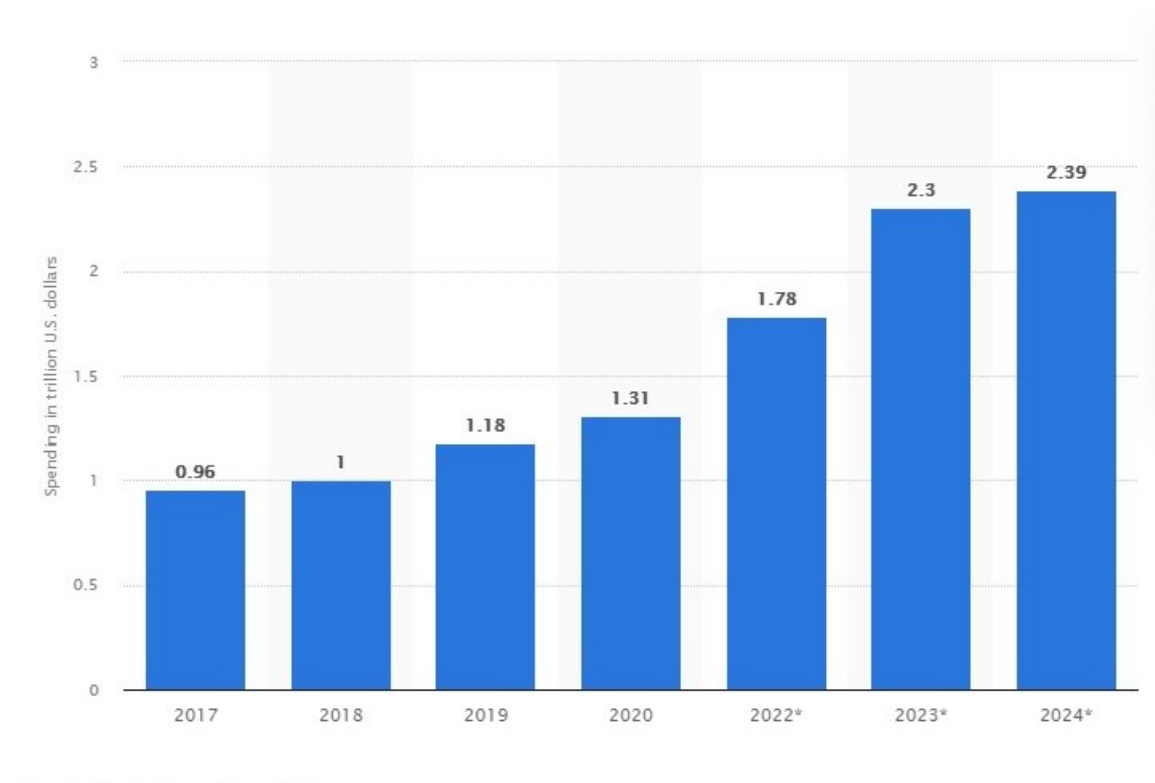


Figura 1.2: Gasto en tecnologías y servicios de transformación digital en todo el mundo de 2017 a 2024. (Fuente: [2])

Este trabajo se enfoca en algunas de las tecnologías emergentes que utilizan las empresas del sector industrial, en concreto, se analizan las industrias de cosmética que emplean líneas de producción para desarrollar sus productos. Una línea de producción se basa en montar estaciones sucesivas de trabajo, donde los procesos se encuentran estáticos y es el producto el que va fluyendo.

La primera idea de línea de producción se remonta al año 1901 cuando Ransom Olds, un pionero de la industria automovilística estadounidense, desarrolló el primer prototipo de cadena de montaje denominado Curved Dash para la producción de automóviles en masa. Tomó la idea del concepto de *taylorismo*, método de organización industrial que se basa en la división de tareas en el proceso productivo para aumentar la productividad. Pero, no fue hasta 1908 cuando el concepto de línea de producción tomó popularidad con el emprendedor Henry Ford, quien comenzó la producción de automóviles a gran escala.

Los parámetros más significativos a tener en cuenta dentro de una línea de producción son los siguientes:

- **Tiempo de ciclo (TC):** Indica cada cuanto tiempo sale una pieza por final de línea, o, lo que es lo mismo, el tiempo de la estación más lenta y que, por tanto, marca el ritmo de producción. Se

mide en *minutos/pieza* (ver 1.1).

- **Capacidad de producción (Cap):** Indica el número de piezas por unidad de tiempo que puede producir una estación de trabajo. Se mide en *piezas/hora*.
- **Tiempo o Work Estándar (WS):** Indica el tiempo de operario (tiempo visto como recurso) necesarios para fabricar una pieza. Se mide en *minutos de hombre/pieza* (ver 1.2).
- **Productividad (P):** Se mide en términos de **OUTPUT** (resultados obtenidos)/**INPUT** (resultados empleados). Es útil expresar la productividad en piezas que salen de la línea entre los recursos humanos de los que se dispone *piezas/horas—por—hombre* (ver 1.3).

$$TC = \frac{1}{Cap} \quad (1.1)$$

$$WS = \frac{1}{P} \quad (1.2)$$

$$P = \frac{OUTPUT}{INPUT} = \frac{\text{Nº de Productos}}{\text{Nº de horas por hombre}} \quad (1.3)$$

Resulta muy conveniente para las industrias tener en cuenta estos parámetros para poder aumentar la productividad, que, como hemos adelantado al principio de este apartado, es uno de los objetivos fundamentales.

En base a la definición que tiene la productividad en las empresas, podemos llegar a diferentes conclusiones:

1. En una primera aproximación se podría pensar que para aumentar la productividad hace falta disminuir el denominador o, aumentar el numerador. Es decir, fabricar lo mismo pero con menos recursos humanos o, utilizar los mismo recursos pero fabricar más productos:

$$\frac{\text{Nº de Productos}}{\downarrow \text{Nº de horas por hombre}} = \uparrow P \quad (1.4)$$

$$\frac{\uparrow \text{Nº de Productos}}{\text{Nº de horas por hombre}} = \uparrow P \quad (1.5)$$

2. Pero, la mejor opción, y por la que apuestan las industrias, es la de aumentar significativamente los productos utilizando mayor cantidad de recursos, tanto humanos como tecnológicos.

$$\frac{\uparrow\uparrow \text{Nº de Productos}}{\uparrow \text{Nº de Recursos}} = \uparrow P \quad (1.6)$$

Los recursos tecnológicos clave que pueden implementarse en una línea de producción de cosmética son, entre otros muchos, las tecnologías de **Automatización** y de **Trazabilidad**:

Automatización

La Automatización en las líneas de producción tiene como objetivo principal la integración de máquinas y aplicaciones que sustituyan los procesos manuales, para disminuir costes, acelerar el tiempo de ciclo y eliminar los posibles errores humanos que pueden cometerse al trabajar de forma manual [11]. Todo esto resulta en una mayor productividad para la empresa y en un aumento significativo del número de productos, que conlleva, a su vez, mayor beneficio económico y competitividad con otras empresas del mismo sector.

Trazabilidad

El concepto de Trazabilidad se utiliza en la industria para referirse al seguimiento de los productos en toda la cadena de suministro, desde las materias primas con la que se elaboran hasta su llegada a la tienda o empresa final [12]. En concreto, la trazabilidad dentro de la línea de producción, llamada trazabilidad interna o de proceso, ofrece beneficios como asegurar la calidad de los productos, gracias a la rápida identificación de defectos en ellos.

En el capítulo 2 se desarrollan de forma más extensa estos dos conceptos, para tener una visión más amplia de la importancia de implementarlos en las líneas de producción de cosmética. Las industrias de cosmética son uno de los sectores en los que se considera más crítico tener implantado un buen sistema de trazabilidad, tanto interno como externo, debido a que sus productos afectan directamente a la salud y seguridad de los consumidores [13].

1.2 Objetivos

El objetivo fundamental del proyecto es el desarrollo de una aplicación de Pick&Place utilizando el software, de la empresa Omron, Sysmac Studio, aplicando un sistema de trazabilidad del producto, como ejemplo de la maximización de la productividad en una línea de producción de productos cosméticos de ejemplo.

Se creará para ello un robot SCARA simulado, con el paquete de Sysmac Studio 3D, que realice el movimiento de coger y dejar una pieza empleando el módulo específico de Robótica, que incorpora el controlador de máquina NJ1500-4500, para establecer su cinemática. La elección de la aplicación concreta viene debida al deseo de realizar la simulación del robot Delta recientemente donado a la Escuela Politécnica Superior por Omron.

Como se ha comentado en el apartado de motivación, una parte indispensable de una línea de producción es un sistema de trazabilidad capaz de detectar y localizar modelos previamente configurados, para llevar un control del producto dentro de la línea. Por ello es objetivo fundamental de este trabajo mostrar, utilizando el sensor de visión FQ-M y el software Sysmac Studio para su configuración, algunos de los métodos que se utilizan en las industrias cosméticas para la detección de piezas.

Como pasos previos a conseguir los objetivos de este trabajo se han seguido una serie de objetivos parciales que se nombran a continuación:

- Breve estudio del concepto de *Automatización y Trazabilidad* para el incremento de la productividad en las líneas de producción, en concreto en las industrias de cosmética.
- Descripción detallada de los elementos que conforma una solución de Pick&Place de la empresa Omron, que resulta útil para entender algunos conceptos de la programación de las aplicaciones.
- Realización de una breve guía de usuario para resumir el software Sysmac Studio.
- Desarrollo completo de la aplicación de trazabilidad, con el sensor de visión FQ-M, y la aplicación de Pick&Place, creando un robot SCARA 3D.

1.3 Estructura

El proyecto está estructurado en cinco capítulos principales que se explican a continuación:

- **Capítulo 1 - Introducción:** En este apartado se hace una breve descripción del problema planteado, *la importancia de la implementación de soluciones tecnológicas en las empresas para poder aumentar la productividad.*
- **Capítulo 2 - Base Teórica:** En este capítulo se detalla en profundidad las tecnologías claves empleadas en las líneas de producción. Además, se describen los elementos fundamentales que forman parte dentro de una aplicación de Pick&Place (una sección fundamental que tienen todas las líneas de producción). Por último, se desarrolla una breve guía de usuario para entender el programa que se ha utilizado para llevar a cabo la parte práctica de este trabajo.
- **Capítulo 3 - Desarrollo:** Se explican en profundidad las funciones utilizadas para la aplicación de Pick&Place, y los elementos empleados en el software, Sysmac Studio, para la aplicación de Trazabilidad.
- **Capítulo 4 - Resultados:** Se muestran los resultados obtenidos de las aplicaciones realizadas.
- **Capítulo 5 - Conclusiones y trabajos futuros:** Se detallan las conclusiones y se exponen algunas propuestas futuras.

Capítulo 2

Base Teórica

La definición aportada por la Administración de Alimentos y Medicamentos dice que un cosmético es una “sustancia destinada a ser aplicada en el cuerpo humano para limpiar, embellecer o alterar la apariencia sin afectar la estructura del cuerpo o funciones”. Las industrias de cosmética han de encargarse, por lo tanto, de desarrollar dichos productos asegurando siempre la buena calidad de los componentes utilizados.

Esto lo hacen bajo el seguimiento de la Norma ISO (Organización Internacional de Normalización) 22716 publicada en 2007 como una *“guía sobre la fabricación segura de productos de cosmética”*. La Norma ISO 22716 tiene como principal objetivo ayudar a asegurar la calidad del producto final y a proteger al consumidor, lo que supone una gran ventaja para las industrias de cosmética ya que, en los últimos años, las líneas de producción se han vuelto cada vez más extensas y complejas, y ha resultado cada vez más difícil llevar un control y supervisión de la calidad y la seguridad de los artículos [14].

En los últimos años, debido al aumento de la preocupación por la imagen personal, las industrias de cosmética han experimentado un crecimiento exponencial en la venta de sus productos y, por lo tanto, en sus beneficios anuales. Aunque, es un sector que puede presentar grandes pérdidas de hasta 509 mil millones de euros por culpa de falsificaciones y mercados paralelos [15]. La trazabilidad en sus líneas de producción, no solo para cumplir con la Norma ISO 22716 para la buena calidad del producto, sino para cuidar el *packaging* y luchar contra la falsificación, se ha vuelto de vital importancia y resuelta de gran interés para este proyecto.

Para implementar un buen sistema de trazabilidad es necesario contar con líneas de producción automatizadas que puedan tomar decisiones de forma independiente sin necesidad de personas a su cargo. Como ejemplo, se podría pensar en el caso de una línea de producción en la que un sistema de trazabilidad detectara que existen fallos en algunos de sus artículos. Estos, deberían ser inmediatamente retirados de la cinta para mantener la buena calidad de los productos. Contar con líneas automatizadas, donde el sistema de trazabilidad se conecte directamente con un robot autómatas que retire el producto defectuoso cuando es detectado, aumenta la productividad, los beneficios y la competitividad de las industrias.

En las siguientes secciones 2.1 y 2.2 se expondrán los conceptos de *Automatización* y *Trazabilidad*, sus principales características y los beneficios de aplicar sus soluciones en las industrias de cosmética.

2.1 Automatización

En RedHat, se define la automatización como: *"La forma de usar la tecnología para realizar tareas casi sin necesidad de las personas. Se puede implementar en cualquier sector en el que se lleven a cabo tareas repetitivas. Sin embargo, es más común en aquellos relacionados con la fabricación, la robótica y los automóviles, así como en el mundo de la tecnología, como el software de decisiones empresariales y los sistemas de TI."*

En otras palabras, se podría definir la **Automatización industrial** como el empleo de diferentes tecnologías para controlar y monitorizar procesos, máquinas y dispositivos que realizan tareas repetitivas minimizando la intervención humana.

En la historia, la automatización se implementó la primera vez que se construyeron objetos que imitaban las partes del cuerpo humano. Estas primeras creaciones datan de la época de los egipcios y griegos, quienes construyeron brazos y sistemas hidráulicos con motivos de oración a los dioses. Los primeros autómatas mecánicos no llegaron hasta mediados del siglo XVIII, a manos de Jacques de Vaucansons, seguido de Henri Maillardet, quienes los crearon con fines lúdicos. El mayor avance en la automatización y, sobre todo, en la automatización industrial, se produjo con la Revolución Industrial a principios del siglo XX. El uso de los semiconductores, permitió crear autómatas cada vez más pequeños, más ágiles y de mayor capacidad de automatización, todo esto hasta nuestro tiempos, donde la automatización está a la orden del día, en especial en los procesos productivos, con la llegada de tecnologías como la robótica, el blockchain, el 5G o el machine learning, entre muchas otras, y que compone lo que se llama ahora la **Industria 4.0 o Cuarta Revolución Industrial** [16].

El termino de Industrial 4.0 fue utilizado por primera vez por el Gobierno alemán que lo describió como una organización de procesos de producción basados en la tecnología y en dispositivos que se comunican de forma autónoma a lo largo de la cadena de valor (*Smit et. al. 2016*).

Las tecnologías en las que se fundamenta la cuarta revolución industrial se han estado utilizando en las industrias los últimos años de una manera aislada. El objetivo es que, para la tercera década del siglo XXI todas las cadenas de valor se transformen en un flujo completamente integrado, automatizado y optimizado, que mejorará la eficiencia y la productividad de la empresa.

A continuación se nombran algunas de estas tecnologías [17]:

- **Big Data:** Este término hace referencia a cantidades de datos con un gran nivel de volumen y complejidad que no pueden ser tratados por los softwares convencionales. A diferencia de los sistemas tradicionales que usaban fuentes de datos estructuradas, el Bid Data obtiene los datos de diversas fuentes, incluyendo tanto datos estructurados como no estructurados, los procesa, los almacenan y los carga de manera distribuida entre varios procesadores.
- **Robots autónomos:** Se les llama a los robots que se han vuelto mucho más autónomos, flexibles y cooperativos ya que poseen características como poder interactuar entre ellos, realizar decisiones operativas por ellos mismos (gracias a la inteligencia artificial) y ejecutar trabajos y actividades por largos periodos de tiempo si la necesidad de la intervención humana.
- **Simulación:** La simulación en 3D tomará una gran importancia en las plantas de producción,

esto permitirá reproducir el mundo físico en un mundo virtual para corregir errores y optimizar la programación antes de ponerlo en práctica.

- **Internet de las cosas (IoT):** Los dispositivos estarán adaptados para poder comunicarse entre ellos a través de internet, lo que permitirá la toma de decisiones en tiempo real.

En la aplicación de Pick&Place que se realiza en el siguiente capítulo, apartado 3.2, se utiliza una herramienta de simulación en 3D para crear un Robot autónomo que recoja y deje una pieza. La utilización de herramientas en 3D en la industria, como se ha comentado anteriormente, facilita la detección y corrección de errores, tanto estéticos como funcionales. A nivel técnico, la simulación 3D puede detectar interferencias entre piezas, calcular el peso de las piezas en función de una densidad determinada o establecer las resistencias de los materiales [18].

Gracias al auge de estas tecnologías han surgido grandes soluciones para estos problemas a manos de importantes empresas como: Omron, ABB o Siemens. Para este proyecto se ha escogido trabajar con la solución de una plataforma de software para la automatización de la empresa Omron.

Para Omron un sistema de automatizado de una línea de producción está compuesta por los elementos que se pueden observar en la figura 2.1.

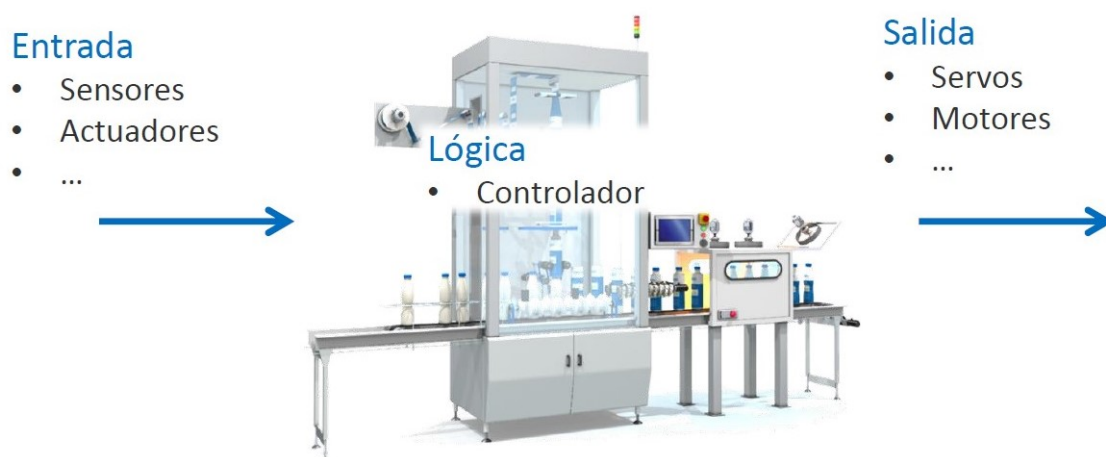


Figura 2.1: Partes de un proceso de automatización.

- **Entrada:** Todo sistema de automatización debe tener una entrada de información como la que proporcionan los sensores, actuadores, cámaras, etc, para recoger todos los datos que entren y poder después tomar decisiones en base a ellos. Estos sistemas de recogida de datos deben ser lo más precisos y rápidos posibles para no incurrir en errores posteriores y obtener la mayor cantidad de datos en tiempo real.
- **Lógica:** Esta parte se encarga de recibir los datos de la entrada y crear las ordenes de ejecución que se enviarán a la salida. En Omron ya no se utilizan controladores lógicos programables (PLC), sino lo que llaman ellos **controladores de máquina** que, a diferencia de los anteriores, poseen la lógica y el motion integrados en el mismo hardware.

- **Salida:** Toma los datos de la lógica de control para ejecutar las ordenes. Está compuesto generalmente por robots, motores, servos, etc. Los robots son instrumentos que pretenden replicar los movimientos humanos para aportar mayor precisión, fuerza y seguridad con el fin conseguir una mayor productividad en la línea de producción.

En el caso de la aplicación de Pick&Place de este proyecto, se quiere imitar como se llevaría acabo la acción de retirar un producto dañado en una industria de cosmética. La entrada correspondería con un sensor de visión que mandaría al controlador la posición de una pieza defectuoso en la línea, el controlador de máquina actuaría y mandaría a la salida, Robot SACARA, la orden de movimiento para coger y retirar dicha pieza.

Para finalizar con este apartado se listan algunas de las ventajas que un sistema de Automatización puede ofrecer a las líneas de producción de la industrias de cosmética [19]:

- Minimizar los fallos gracias a la estabilidad y robustez de las máquinas, y al mínimo empleo de personal humano.
- Reduce costes gracias a la disminución de fallos y a la calidad de los productos.
- Mejora las condiciones de trabajo y reduce los riesgos de accidentes laborales.
- Mejora la eficiencia y productividad.
- Reduce el tiempo de ciclo de fabricación.

2.2 Trazabilidad

La trazabilidad está definida por la Organización Internacional para la Estandarización ISO 9000 como “la capacidad de rastrear la historia, la aplicación o la ubicación de lo que está bajo consideración” ISO (2005).

Según el Comité de Seguridad Alimentaria AECOC se entiende trazabilidad como *el conjunto de aquellos procedimientos preestablecidos y autosuficientes que permiten conocer el histórico, la ubicación y la trayectoria de un producto o lote de productos a lo largo de la cadena de suministros en un momento dado, a través de unas herramientas determinadas.*

En otras palabras, podemos entender la trazabilidad en una empresa como una serie de procesos que permiten identificar un determinado producto, y a su vez conocer su origen, su ubicación y su trayectoria.

El proceso de trazabilidad total de un producto engloba tres tipos de trazabilidad que se enumeran a continuación:

1. Trazabilidad hacia atrás o ascendente (*tracing*).
2. Trazabilidad interna o de proceso.
3. Trazabilidad hacia adelante o descendente (*trakcing*).

Para la aplicación que se ha llevado a cabo, como se ha querido imitar la trazabilidad de los productos en una línea de producción de cosmética, el único tipo que nos interesa es *la trazabilidad interna o de proceso*, aun así, se expondrá brevemente cada una de ellas, ya que es de vital importancia entender la trazabilidad en la industria en su totalidad.

TRAZABILIDAD HACIA ATRÁS (*tracing*):

La trazabilidad hacia atrás proporciona información de los productos que entran en la empresa y quienes son los proveedores de los mismos.

De acuerdo con Kelepouris et al. (2007), se dice que dependiendo de la dirección en el cual la información es recordada en la cadena, trazabilidad hacia atrás o *tracing es la capacidad, en cada punto de la cadena de suministro, de encontrar el origen y las características de un producto en base a uno o varios criterios dados*.

Otra definición de la trazabilidad hacia atrás es la estipulada por Manos & Manikas (2010), los cuales la definen como *la capacidad de rastrear la historia del producto a través de la cadena de suministro desde o hacia el lugar y hora de producción, incluyendo la identificación de los productos utilizados y la producción y operaciones llevadas a cabo*. Ver Figura 2.2.

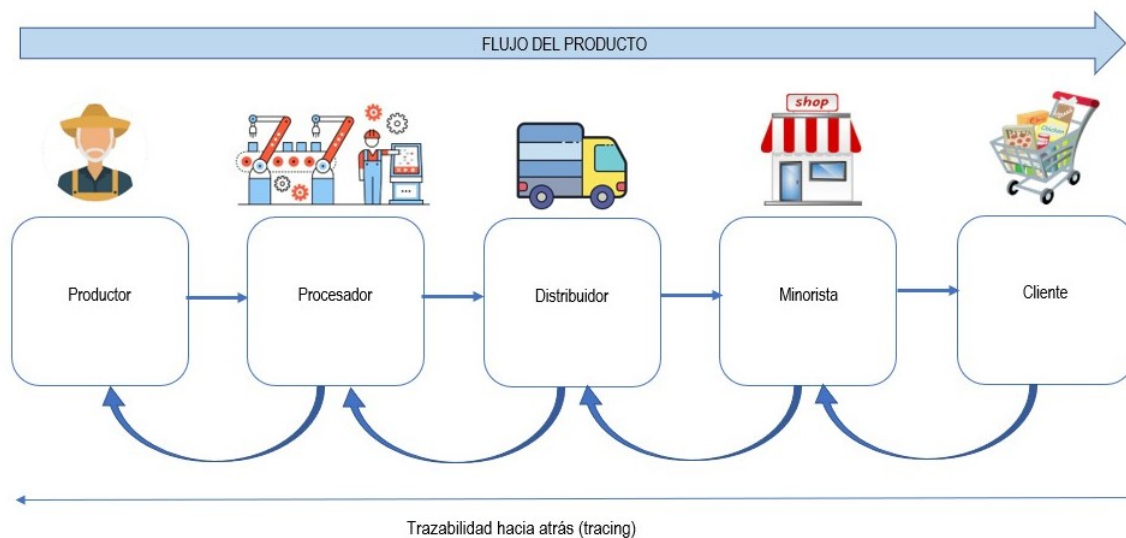


Figura 2.2: Trazabilidad hacia atrás en dirección al actor implicado, “aguas arriba”.

TRAZABILIDAD INTERNA O DE PROCESO:

La trazabilidad interna consiste en la trazabilidad del producto a lo largo de la cadena productiva de la empresa, industria o sector, desde la recepción de las materias primas hasta la expedición del producto terminado.

La trazabilidad interna desempeña un papel muy importante en la optimización de la cadena de suministro y aporta grandes ventajas como [20]:

- **Mejora la calidad de los procesos:** a medida que se documentan sus procesos y servicios, es posible identificar fácilmente errores y procedimientos incorrectos. Como resultado, la empresa puede planificar objetivos y acciones más específicos, manteniendo la seguridad y la eficiencia de su gestión.
- **Minimiza costos:** Minimiza costos: esta es una consecuencia directa de la ventaja anterior, ya que, con la mejora de la cadena de producción, evitando los fallos que ocurren en cada operación y las pérdidas involucradas, los costos de los procesos internos se reducen considerablemente.
- **Previene recurrencias:** durante la cadena de producción, los bienes pueden presentar problemas de diferentes causas, para evitar que alguna de ellas se repita en el futuro, es necesario que la empresa comprenda cómo ocurrió y cuál es exactamente el error que debe corregirse.

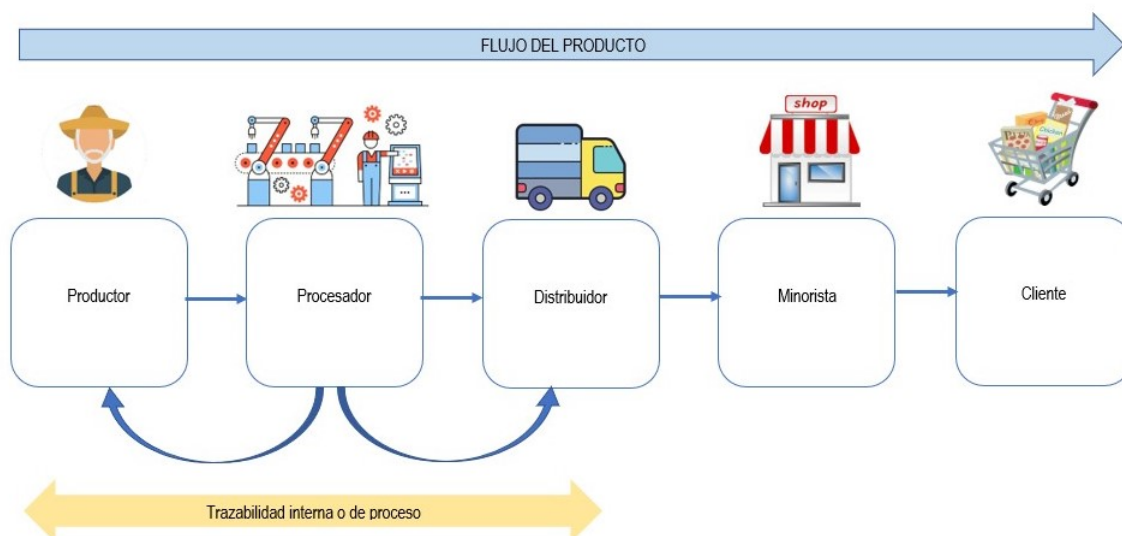


Figura 2.3: Trazabilidad interna o de proceso.

TRAZABILIDAD HACIA DELANTE (*traking*):

Según distintos autores este tipo de trazabilidad tiene el objetivo de rastrear, es decir, localizar la entidad (lote, pedido, componente, etc.). De acuerdo con Schawagele (2005) la trazabilidad hacia adelante puede definirse como *la capacidad de seguir el camino de un elemento a medida que se mueve aguas abajo (a partir del actor implicado) a través de la cadena de suministro desde el principio hasta el final*". Ver figura 2.4.

Por lo tanto, se puede decir que la trazabilidad hacia adelante se aplica a los productos que salen de la empresa, teniendo en cuenta los destinos a los que van destinados.

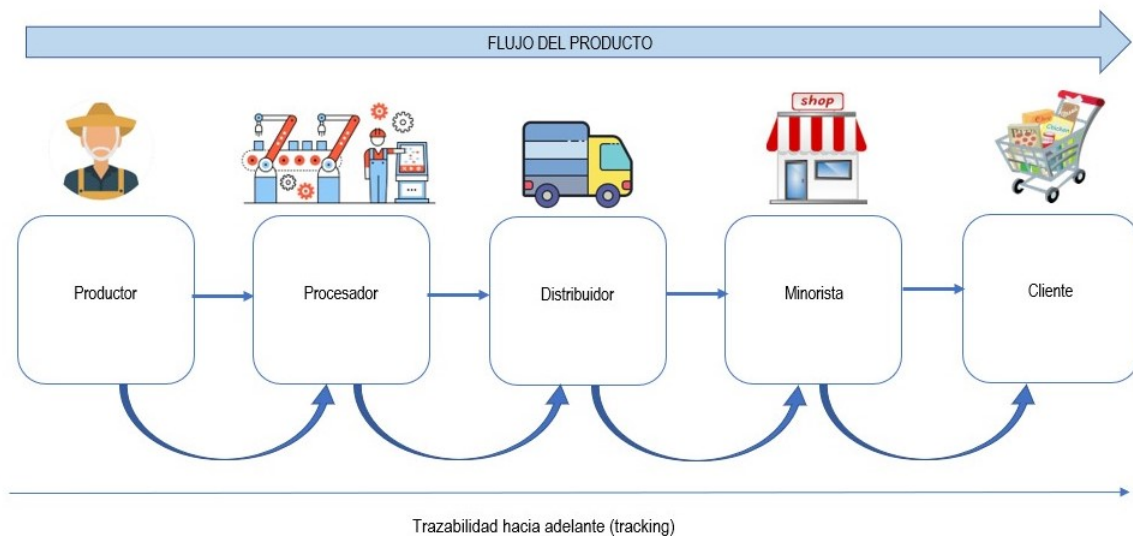


Figura 2.4: Trazabilidad hacia adelante a partir del actor implicado, “aguas abajo”.

ELEMENTO QUE COMPONEN UN SISTEMA DE TRAZABILIDAD

Para que un sistema de trazabilidad sea efectivo debe estar formado por tres elementos indispensables que se describen a continuación [21]:

- **Sistema de identificación:** Son los encargados de llevar a cabo la identificación del producto. Aunque en el mercado existan diversos tipos de sistemas de identificación, que aportan diferentes soluciones, es deber de la empresa escoger el que mejor se adapte a sus necesidades. Más adelante se explicaran, lo que se han considerado como, los sistemas de Identificación más interesantes y utilizamos actualmente por las empresas.
- **Sistema de captura de datos:** Son también de vital importancia a la hora de implementar un buen sistema de trazabilidad. Los sistemas de captura de datos, como su propio nombre indica, tienen el propósito de capturar datos en un tiempo o momento determinado de la cadena de suministro. Esto es interesante, ya que se puede conocer y almacenar información referente al producto a seguir de forma progresiva, es decir, desde las materias primas utilizadas, pasando por el pesado y hasta los datos de distribución. Es por ello, que será necesario la utilización de herramientas y equipos capaces de realizar estas labores (antenas RFID, lectores de códigos de barras, sensores de temperatura y humedad, entre otros).
- **Sistema para la gestión de datos:** Una vez que los productos son identificados y sus datos son capturados correctamente es esencial poder analizarlos y darles un uso adecuado, es por ello, que es fundamental utilizar sistemas para la gestión y administración de datos como softwares especializados que incluyan funciones como la impresión de etiquetas o códigos de barras, al igual que la opción de grabar chips de tipo RFID, entre otros.

Como se adelantó en la introducción de este trabajo, apartado 1, un correcto uso de un sistema de trazabilidad puede llevar a la empresa a su mayor objetivo, el incremento de la productividad y la

eficiencia en sus líneas de fabricación. Pero, para ello, la empresa debe conocer bien sus necesidades y capacidades, y escoger el sistema de identificación ideal para sacar la máxima rentabilidad a esta tecnología. Los sistemas de identificación que más se utilizan en las líneas de producción actualmente se listan a continuación:

1. **Sistema de visión:** Es la utilización de cámaras de vídeo o imágenes fotoeléctricas conectadas a un software que permite identificar productos a través de sus características, formas o especificaciones.
2. **Código de barras:** Es un sistema compuesto por líneas y espacios que sirve para la captura de información de los productos.
3. **Sistema de radiofrecuencia:** Son sistemas que utilizan el principio de ondas de radio para identificar los productos y tiene la capacidad de almacenar y actualizar datos durante toda la cadena de suministro.

Para una industria de cosmética, el uso de un sistema de visión puede ser altamente beneficioso para realizar todas las comprobaciones de calidad, como, por ejemplo, roturas, o cantidad de envasado, y poder desechar los productos que no cumplen con un juicio determinado. Por otro lado, el código de barras, se suele utilizar para controlar la cantidad de productos que entran y salen de las líneas, conocer información sobre su origen y destino, y, en general, tener un control individual de cada producto. Los sistemas de radiofrecuencia, aunque son más caros que los códigos de barras, ofrecen una mayor durabilidad y mayor procesamiento de información, que resulta muy beneficioso en este sector, ya que se pueden guardar más datos como el peso de los productos y, la cantidad y calidad de los componente utilizados.

Para la aplicación de trazabilidad que se desarrolla en la sección 4.1 se utilizará únicamente el sistema de visión, aun así, a continuación, se desarrollan de forma más extendida cada uno de los sistemas nombrados anteriormente, analizando su arquitectura, sus ventajas y desventajas y alguno de sus usos más comunes.

2.2.1 Visión artificial

La visión artificial es una disciplina científica que consiste en la captura y el procesamiento de imágenes del mundo real, para que puedan ser procesadas por una máquina, ordenador o sistema digital. La analogía de esta tecnología, sería la misma que nosotros realizamos con los ojos, es decir, percibimos imágenes del mundo que nos rodea, para tomar decisiones en base a ellas.

Arquitectura

En la siguiente Figura 2.5, podemos ver un ejemplo de cámara de visión artificial, que, a priori, puede ser igual que una cámara tradicional, pero que incluye unas ciertas diferencias. Principalmente este tipo de cámaras poseen una mayor capacidad de procesamiento, pues no solo tienen que procesar la captura de la imagen, si no interpretar qué es lo que están capturando. Es por ello, incluyen procesadores más potentes que una cámara tradicional.



Figura 2.5: Ejemplo de cámara de visión artificial, en aplicaciones de trazabilidad industrial. (Fuente: [3])

Usos más comunes

Una de las principales aplicaciones de un sensor de visión artificial es en un sistema de Pick&Place, cuya aplicación se verá más adelante en este proyecto. A continuación, se explican los elementos básicos de los que consta, y la importancia de la visión artificial en este sistema.

- **Cámara de visión artificial:** Todo sistema de este tipo necesita una cámara por delante del Robot para detectar todos los productos que se mueven por la línea. En base a un modelo del producto, previamente preestablecido, el sensor de visión detecta si alguno de los productos de las cinta no cumple con las características del modelo y manda la información de su posición para que posteriormente el Robot las elimine de la cinta.
- **Cinta transportadora:** La cinta es necesaria para hacer mover todos los objetos que quieren ser detectados de forma automática. Generalmente, existe una cinta por la que discurren todos los productos a detectar, y es el robot el que los transporta hacia otras cintas para desecharlos.
- **Robot:** Es el encargado de recibir la información de la cámara de visión, en concreto recibe las posiciones de los productos que deben ser retirados de la cinta.

Aunque nos interesa el uso del sensor de visión en la industria de cosmética, este tiene gran cantidad de aplicaciones fuera de ella, desde el reconocimiento facial hasta la conducción autónoma de vehículos, que en los últimos años ha ganado gran popularidad.

A continuación, se enumeran algunas de las ventajas y desventajas de este sistema [3].

Ventajas:

- Permiten llegar donde el ojo humano no puede.
- Aumenta la seguridad y la calidad de los procesos al eliminar parte de la acción humana.
- Genera datos que pueden ser interpretados para la toma de decisiones en tiempo real.
- Mejorar la productividad, y disminuye los costes de producción.

Desventajas:

- Requiere de personal cualificado para su uso.
- Puede suponer un coste mayor que otros sistemas de identificación.

Las ventajas de utilizar la visión artificial en las líneas de producción supera con éxito a las desventajas, ya que el gasto inicial que pueda suponer su compra, instalación y personal, no es comparable con el aumento en la productividad y, por ende, en las ganancias económicas que el uso del sensor de visión conllevan.

2.2.2 Código de barras

El código de barras es un patrón leíble por una máquina que se aplica a los productos, empaques o piezas. Estos, proporcionan información tal como el origen, el destino, el tipo de producto, la información de facturación, entre otros aspectos claves de identificación de producto [4]. Los códigos se leen utilizando un lector especial o un lector de luces y lentes que decodifica la información, que más tarde es transferida a una base de datos donde es ingresada y rastreada.

Hoy en día los códigos de barras presentan diferentes formatos, desde los llamados código de barras unidimensionales o 1D que solo contienen datos en la dirección horizontal, hasta códigos más avanzados llamados bidimensionales o 2D, que están formados por puntos y cuadrados, y permiten almacenar mayor cantidad de datos gracias a contener información tanto en la dirección horizontal como vertical, 2.7.



Figura 2.6: Ejemplo de códigos de barras 1D y 2D. (Fuente: [4])

Arquitectura

Un sistema de código de barras suele estar compuesto por un software (S) y un hardware (H), los cuales son indispensables para su adecuado funcionamiento. En el hardware se incluye el código, los métodos de impresión y marcado, y el lector. El software únicamente comprende el sistema de información que recupera y registra la información detectada por el lector.

Para explicar la función de cada componente en detalle, se describirán a continuación sus generalidades:

- **El código:** Como se ha adelantado anteriormente existen, en general, dos tipos de códigos de barras. Los códigos 1D fueron los primeros que se implementaron a nivel mundial, solo contienen datos en la dirección horizontal y se representan con un conjunto de líneas de diferente grosor y espaciado. Además, contiene un patrón al principio y al final para indicarle al lector donde comienza y termina el código. Debido a la pequeña cantidad de datos que podía almacenar los códigos anteriores, y a la gran cantidad de información que es necesaria almacenar en una cadena de suministro se crearon los códigos 2D (bidimensionales) que contienen información tanto en forma horizontal como vertical, lo que les permite tener mucha más capacidad de almacenamiento. El *GS1 International* ha aprobado seis tipos de códigos de barras, que varían según la aplicación y

la cantidad de información a almacenar. En la Figura 2.7 se muestran dichos códigos, con sus características y distintas aplicaciones. Debido a su gran variedad y las distintas restricciones que presentan, una empresa debe analizar en qué y dónde los utilizará, para seleccionar el código con el que se aprovechen al máximo los beneficios y se garantice su adecuada operación.

EAN(European Article Number)
<ul style="list-style-type: none"> · Usado en retail o tiendas minoristas. · Los más comunes son el EAN-13 y el EAN-8.
RSS (Reduced Space Symbology).
<ul style="list-style-type: none"> · Usado para productos de pequeño tamaño o difícil de marcar. · Puede ser leído en punto de venta. · Menor tamaño que el EAN. · Facilita la trazabilidad y autenticación de productos.
ITF-14
<ul style="list-style-type: none"> · Usado solamente para cajas de cartón corrugado. · No puede usarse en el punto de venta.
GS1 DataBar
<ul style="list-style-type: none"> · Escaneable en puntos de venta. · Menor tamaño que el EAN. · Aplicaciones sector salud.
GS1-128 o EAN-128
<ul style="list-style-type: none"> · Puede contener todas las características y atributos de identificación del producto. · No puede usarse en el punto de venta. · Uso para almacenamiento y control de inventario.
GS1 DataMatrix
<ul style="list-style-type: none"> · Es el único símbolo de matriz 2D. · Ampliamente utilizado en el sector salud.

Figura 2.7: Tipos de códigos de barras.

- **La impresión** de los códigos generalmente se realiza de dos maneras: aplicando el código a empaques o etiquetas, generalmente usando métodos térmicos o de inyección de tinta, o, usando métodos de marcado directo para marcar permanentemente el código directamente en la pieza utilizando impresión por puntos, grabado químico o marcado láser. La impresión de tinta y térmica se utilizan generalmente para imprimir los códigos de barras 1D. Ambas tecnologías se utilizan para imprimir el código sobre un empaque, etiqueta u otro material. El marcado directo de piezas ofrece una solución duradera en comparación a los métodos de impresión, además, suelen incluir más información por lo que se utilizan códigos 2D en lugar de códigos de barras 1D de menor ancho de banda. Entre los más utilizados está el sistema de marcado láser que suele utilizar láser de fibra para grabar códigos, Data Matrix u otras simbologías 2D sobre la pieza. Los sistemas de marcado de impresión de puntos que suele ser la opción más considerada y más rentable, en ella se utiliza una aguja oscilante para presionar el metal y crear la marca. Y por último, el grabado químico que utiliza una solución basada en sodio combinada con pulsos de corriente eléctrica de bajo voltaje. La solución cargada disuelve el metal que, más tarde, se extrae mediante una plantilla especial [22].
- **Lectores** En [23] se define un lector de código de barras como un dispositivo con luces, lentes y un sensor que decodifica y captura la información contenida en los códigos de barras. Estos lectores utilizan un haz de láser como fuente de luz y espejos oscilantes o prismas giratorios para escanear el haz del láser hacia adelante y hacia atrás del código de barras. Desde el fotodiodo se crea una señal analógica que, luego, se convierte en una señal digital. Uno de los principales beneficios es que

no necesitan de un procesamiento de imagen. Esto hace que sean más rápidos y que puedan realizar hasta 1,300 lecturas por segundo. Gracias al láser pueden leer a largas distancias los códigos de barras 1D haciendo uso de ópticas especiales. Por otro lado, una de sus principales limitaciones es la imposibilidad de leer códigos 2D, que a su vez están aumentando su uso. Además, otro de los problemas es la dificultad que tienen para leer los códigos 1D que poseen bajo contraste, una mala distorsión o que se encuentran dañados, lo que supone un inconveniente añadido pues, las condiciones de trabajo nunca son perfectas lo que supondrá una cantidad considerable de lecturas incorrectas y una ineficiencia en el proceso de producción. Existe, además, otro tipo de lectores de códigos de barras, los basados en imágenes. Estos lectores basan su funcionamiento en la toma de imágenes, utilizando un sensor de matriz de área, similar a los que poseen las cámaras digitales o los móviles. El procedimiento consiste en adquirir una fotografía de los códigos de barras 1D y 2D, para que, después, un software especial de procesamiento de imagen obtenga el código capturado, y envíe los datos a través de la red. Este tipo de sensores tienen un gran beneficio cuando se trata de la lectura de códigos 2D o de matrices de datos, puesto que son capaces de leer varios códigos de barras simultáneos. Por otro lado, no se recomienda su uso para códigos 1D, debido a que son más lentos que los lectores láser, y ello puede suponer una pérdida de rendimiento.

- **Software de información** Los sistemas de información son alimentados por los datos contenidos en el código de barras que son capturados por medio del escáner, el cual minimiza los errores de digitalización. Además, estos sistemas permiten el almacenamiento y análisis de los datos logísticos recopilados.

Usos más comunes

El código de barras es una "Tecnología de Identificación Automática" muy común en el entorno empresarial, con diversas aplicaciones que afectan el proceso de la cadena de suministro, desde las compras, pasando por la preparación y producción de pedidos hasta la gestión de almacenes, control de inventarios, distribución y transporte.

Su uso más común y extendido es en el uso de productos de venta al público, pues permite saber en todo momento de qué producto se trata y cual es su valor.

A continuación, se enumeran algunas de las ventajas y desventajas de este sistema.

Ventajas:

- Reducción en los tiempos productivos de entrada y salida de stock, de reposición, etc.
- Disminución de fallos humanos.
- Impresión de etiquetas de bajo costo.

Desventajas:

- Posible borrado de etiqueta haciendo difícil su identificación.
- Necesidad de impresoras especiales.

- Necesidad de una visión directa con el lector.

El código de barras está implementado en nuestro día a día como una solución ágil y sencilla para controlar ciertos procesos productivos. Claramente las ventajas son mayores que las desventajas, pues su uso está completamente instalado y no parece reemplazable a corto y medio plazo.

2.2.3 RFID

Se trata de una tecnología de identificación por radiofrecuencia, o como sus siglas indican en inglés, "Radio Frequency Identification". Es un sistema que permite la comunicación entre dos o más sistemas, principalmente entre un lector y una etiqueta. La etiqueta permite almacenar datos, para su posterior recuperación por parte del lector.

La etiqueta por norma general, se compone de un microchip y una antena de un tamaño reducido, que permite adherirse a cualquier objeto o superficie. Dicho microchip, suele contener una memoria EEPROM, que contiene la información, y puede ser grabada de fábrica o directamente por el usuario que lo utilice. En la Figura 2.8 se muestra un ejemplo de etiqueta de seguridad, muy usadas en productos de logística para evitar su robo. En el momento de su compra, la información que guarda la etiqueta cambia, y al salir de la tienda, se detecta como producto comprado.

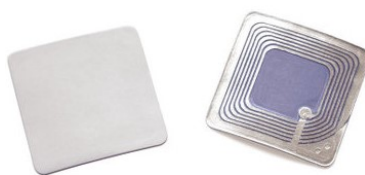


Figura 2.8: Ejemplo de etiqueta RFID. (Fuente: Century Europe)

Las frecuencias utilizadas en este sistema pueden variar según el uso que se les de, en función del alcance, la seguridad o la velocidad de lectura. Por ello, el rango de frecuencias varía entre 50KHz y 2.5GHz. Este rango de frecuencia se recogen en varios estándar ISO, como pueden ser la ISO 14443 o ISO 15692. Según el rango de las frecuencias utilizadas los dispositivos de RFID pueden dividirse en dos grupos:

- Rango comprendido entre 50KHz y 14MHz: tienen un menor coste y son más resistentes al ruido, pero tienen un corto alcance.
- Rango comprendido entre 14MHz y 2.5GHz: tienen un mayor alcance a costa de un mayor costo y tecnología más compleja.

Arquitectura

La arquitectura de un sistema RFID consta de tres elementos: un lector o transceptor, una etiqueta o transpondedor y un subsistema de procesamiento de datos [5]. En la figura 2.9, se puede observar de manera gráfica como funciona esta tecnología.

- **La etiqueta:** está compuesta por una antena y un microchip que contiene la información. La antena tiene el propósito de permitir el envío de la información del microchip al lector. Existen tres tipos

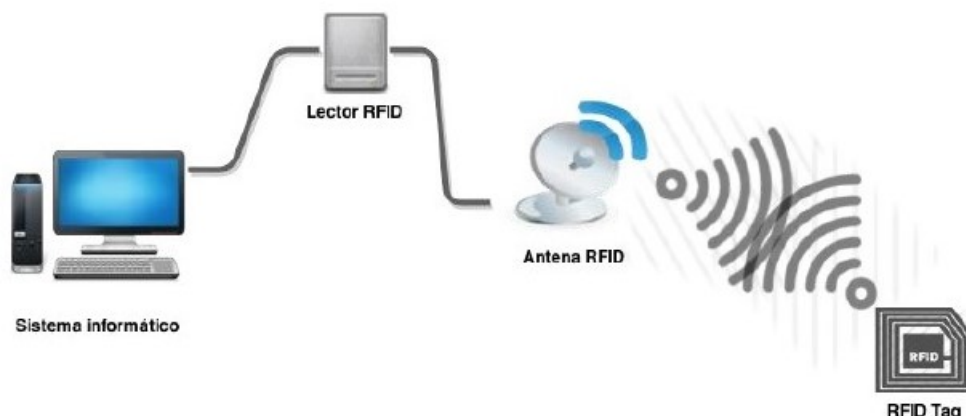


Figura 2.9: Elementos de un sistema RFID. (Fuente: [5])

de etiquetas según su alimentación: **Activas**, si incorporan una batería externa para ampliar y emitir la señal. **Pasivas**, si, por el contrario, no tienen fuente de alimentación externa, emitiendo una señal más débil, pero haciendo posible un tamaño de etiqueta más pequeño. Y, por último, las **Semipasivas**, que incluyen una batería externa, pero es el lector el que emite una señal para obtener la información como en el caso de las pasivas.

- **El lector:** está formado por una antena, un transceptor y un decodificador. Su función consta de emitir continuamente señales para comprobar si hay alguna etiqueta en su radio de emisión, en el caso de encontrar una, obtiene sus datos, los decodifica y se los envía al subsistema de procesamiento de datos.
- **El subsistema:** Es el encargado de procesar y almacenar todos los datos obtenidos del lector.

Usos más comunes

El uso de la tecnología RFID se ha extendido a una gran cantidad de sectores tales como la sanidad, la ganadería, los comercios, pero, sin más importante es en las industrias de logística, farmacéutica y química.

El uso del RFID en la industria permite identificar, rastrear y gestionar productos de una forma fácil y ágil, sin la intervención humana y desde la distancia. Gracias a una mayor automatización, el trabajo de los operarios puede verse aliviado. Además, gracias a la información única recogida en cada chip los productos iguales pueden tener un trato totalmente diferenciado y personalizado, que permita gestionar y almacenar cada producto de forma independiente. Los registros de entrada y salida, es decir, la gestión del stock, también se puede realizar de forma automática gracias al uso de esta tecnología [24].

Existe un aumento notable en el uso de la tecnología RFID que se debe principalmente a la constante necesidad de automatizar los procesos para ganar eficiencia y productividad en las industrias.

A continuación, se enumeran algunas de las ventajas y desventajas del uso de esta tecnología [5].

Ventajas:

- El RFID utiliza radiofrecuencias, lo que es una ventaja frente a infrarrojos, pues el lector y etiqueta

no necesitan estar en visión directa. Esto supone una ventaja frente al código de barras.

- Frente al código de barras, existe otra ventaja, como una mayor durabilidad, puesto que sufre menor desgaste.
- Ofrece lecturas más rápidas y precisas.
- Mejora en la identificación de productos, ya que estos pueden ser iguales, pero tener diferentes códigos.
- A su vez, permite conocer el estado del producto y en que lugar se encuentra, para, por ejemplo, sustraerlo del lugar de almacenamiento.
- Gran capacidad de agilización y automatización de procesos.

Desventajas:

- La principal desventaja es el coste, que es mayor frente al código de barras, cuyo precio puede ser insignificante.
- Corto alcance si no se hace uso de etiquetas activas.
- Las etiquetas activas son grandes y pesadas, ya que tienen incorporada una batería para su funcionamiento.

2.3 Aplicación Pick&Place de Omron

Para aprender en profundidad los elementos indispensables que forman parte en una línea de producción real se va a analizar en detalle cada una de las partes de un Robot demo Delta con aplicación de Pick&Place de la empresa Omron, que se muestra en la figura 2.10. Este Robot es una pequeña aproximación a lo que sería una línea de producción real, pero adaptado para la enseñanza de sus distintos componentes y de su funcionamiento global.

A continuación, se enumeran los diferentes elementos, cuyo índice se corresponde con cada número referenciado en la Figura 2.10. Dentro de la explicación de cada elemento se mostrará una imagen más detallada del elemento en concreto.

1. PANTALLA NS

En la Figura 2.11 se muestra un tipo de pantalla NS fabricada por Omron. Se trata de una interfaz HMI que se utiliza para conectar al humano con la máquina. Consta de una pantalla TFT (Thin Film Transistor) que aporta mayor calidad de imagen. Aunque permite realizar una programación sencilla el principal inconveniente de este tipo de pantalla es que, al ser antigua, no es posible programarla con el software Sysmac Studio, y su comunicación se hace vía Ethernet/IP.

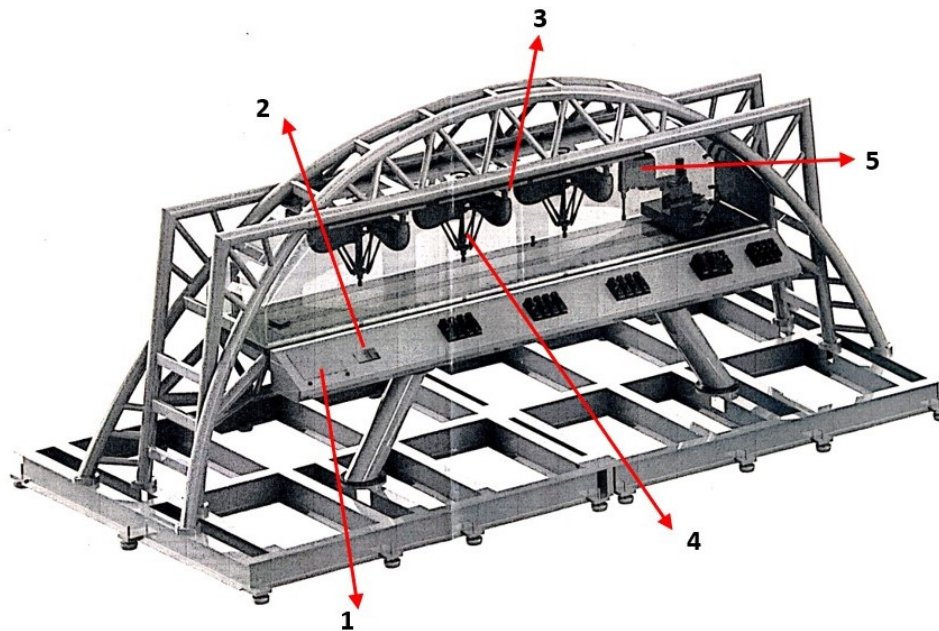


Figura 2.10: Robot Demo aplicación Pick&Place de la empresa Omron.



Figura 2.11: Pantalla NS, modelo NS15/NS12/NS10/NS8 (Fuente: [6]).

2. CONTROLADOR DE MÁQUINA NJ501-4500

Los controladores de máquina NJ de Omron se diferencian de los PLCs convencionales en que la lógica y el motion están integrados en el mismo hardware, y su ejecución puede darse en la misma tarea. Existen 11 modelos de controlador NJ5, que se diferencian en el número de ejes y en los módulos específicos que incorporan. Todos utilizan el procesador Intel Atom a 1.6GHz y un ciclo mínimo de $0.5\mu s$. Tienen los puertos Ethernet/IP y EtherCAT integrados para comunicarse con distintos dispositivos.

Todos los controladores de máquina llevan siempre cuatro módulos integrados:

- El **módulo de función PLC** ejecuta el programa de usuario, envía los comandos al módulo de *Motion Control* y proporciona una interfaz a USB y SD.
- El **módulo de Motion Control** que ejecuta los procesos de *Motion* basados en valores objetivos (posición, velocidad deseadas...) de las instrucciones motion del programa de usuario. Este módulo recibe y envía la información a través del módulo maestro de EtherCAT.

- El **módulo de función maestro de EtherCAT** se encarga de comunicar con los esclavos de EtherCAT
- El **módulo de función de EtherNET/IP** se encarga de realizar las comunicaciones EtherNet/IP.

En concreto, el controlador NJ501-4500, figura 2.12, pertenece al módulo NJ5 Robotics, lo que quiere decir que, a parte de los cuatro módulos imprescindibles, presenta un **Módulo de Robótica** que lleva implementado directamente las funciones de robótica, lo que hace más fácil la programación con Sysmac Studio de los Robots Delta, Scara, etc. El número *4500* indica que puede sincronizarse como máximo con 64 ejes reales o virtuales.

Presenta dos modos de operación. Cuando está en modo *RUN* únicamente se ejecuta el programa de usuario y las variables pueden ser monitorizadas. El modo *PROGRAM* se utiliza para transferir el proyecto y comprobar el cableado de entradas y salidas, pero no se ejecuta el programa de usuario.



Figura 2.12: Controlador de máquina NJ501 - 4500. (Fuente:[6])

3. SENSOR DE VISIÓN ARTIFICIAL FQ-M

El sensor de visión FQ-M, que se observa en la figura 2.13, fue desarrollado por Omron y está pensado específicamente para aplicaciones de Pick&Place, ya que tiene una entrada de encoder para permitir la función de seguimiento (*tracking*), además de ser un dispositivo compacto y rápido.



Figura 2.13: Sensor de visión FQ-M. (Fuente:[6])

Sus principales características son [6]:

- En la mayoría de aplicaciones el sensor de visión se complementa con otros dispositivos (controladores, robots, servos, encoders...) y lo hace a través de EtherCAT o Ethernet estándar.
- Permite detectar hasta 32 piezas simultáneamente en un único disparo, y más de 5.000 piezas por minuto.
- Tiene la capacidad de procesar la imagen recogida y de acuerdo al programa y juicio definido determinar qué datos manda a los periféricos.
- Ofrece una función de seguimiento en tiempo real, ya que el sensor envía las coordenadas de posición de los objetos detectados y es capaz de almacenarlos en una cola permitiendo que las coordenadas de un objeto no se envíen más de una vez.
- Se puede configurar mediante el software de Sysmac Studio de Omron o a través de una consola TouchFinder. En el apartado 4.1 se muestra una aplicación real utilizando el sensor de visión FQ-M y el software de Omron Sysmac Studio.

Con respecto a la imagen, el sensor de visión FQ-M utiliza lentes de tipo C estándar, lo que hace que se adapta bien al campo de visión, a la distancia y a cualquier situación especial, además, permite la instalación de filtros.

Utilizar un sensor de visión en una aplicación de Pick&Place ofrece grandes ventajas en la industria:

- Son más rápidos que las personas por lo que aumenta la capacidad de producción.
- En ocasiones pueden ser más fiables que el ser humano por lo que se pueden evitar errores.
- Se puede combinar con inspección midiendo la calidad del producto.
- Mejora la calidad del producto ya que no existe manipulación directa del mismo.

ROBOTS INDUSTRIALES

Un robot industrial, según la Asociación de Industrias de Robótica (RIA, *Robotic Industry Association*) es un manipulador multifuncional reprogramable, capaz de mover materias, piezas, herramientas, o dispositivos especiales, según trayectorias variables, programadas para realizar tareas diversas. Mientras que, la Federación Internacional de Robótica (IFR, *International Federation of Robotics*) declara que *por robot industrial de manipulación se entiende una máquina de manipulación automática, reprogramable y multifuncional con tres o más ejes que pueden posicionar y orientar materias, piezas, herramientas o dispositivos especiales para la ejecución de trabajos diversos en las diferentes etapas de la producción industrial, ya sea en una posición fija o en movimiento.*

Los robots industriales están formados por eslabones, estructuras rígidas, que se unen entre sí mediante articulaciones para permitir el movimiento. A todo el conjunto de eslabones y articulaciones se le denomina **cadena cinemática**. Cuando la cadena cinemática es abierta los eslabones se encuentran conectados de forma secuencial, desde la base hasta el actuador final. Sin embargo, cuando la cadena cinemática es cerrada los robots presentan elementos en paralelo [7].

Dependiendo de la cinemática que presenten los robots se pueden clasificar en dos tipos, figura 2.14:

- a. **Robot con estructura tipo serie:** Presentan una cadena cinemática abierta, con una estructura similar al brazo humano (antropomórficos).
- b. **Robot con estructura de tipo paralelo:** Presentan una cadena cinemática cerrada, ya que están formados por dos plataformas, una fija y otra móvil, unidos por varios brazos en paralelo.

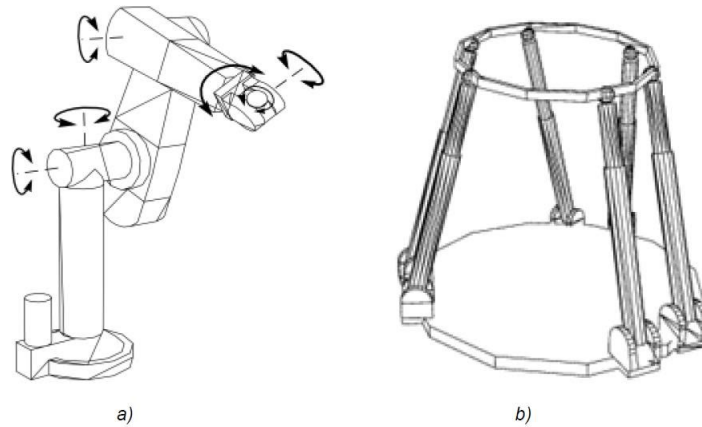


Figura 2.14: Estructura cinemática de un robot, a) serie, b) paralelo (Fuente [7])

En los últimos años, los robots de tipo paralelo están sustituyendo a los de tipo serie por sus grandes ventajas. Los robots de tipo paralelo admiten mayores aceleraciones y velocidades durante su movimiento, además, tienen mayor precisión y rigidez ya que soportan la carga por medio de varios brazos posicionados en paralelo. Aunque, como inconveniente, tiene menor espacio de trabajo [25].

Para entender los robots industriales en su totalidad es necesario conocer los parámetros que los caracterizan:

- Grados de libertad: Se le llama a cada una de las coordenadas independientes que son necesarias para describir el estado del sistema mecánico del robot (posición y orientación en el espacio de sus elementos) [7].
- Espacio de trabajo: Es el conjunto de puntos en el espacio al que el punto terminal tiene acceso.
- Capacidad de posicionamiento del punto terminal: Mide el grado de exactitud que tiene un robot al realizar un movimiento previamente programado.
- Capacidad de carga: El rango de peso que puede transportar el elemento terminal del robot.
- Velocidad: máxima velocidad que puede alcanzar el robot.

4. ROBOT DELTA

Los robots Delta han tenido un gran auge los últimos años gracias a las ventajas que ofrecen por tener una estructura de tipo paralelo. Están compuestos por dos plataformas en paralelo, una fija y otra móvil de menor dimensión. Ambas bases están unidas por medio de brazos paralelos controlados por servomotores para generar el movimiento.



Figura 2.15: Robot X-Delta 3+1. (Fuente:[6])

Algunas de las ventajas que ofrecen frente a otros robots industriales y a su uso en las industrias para automatizar procesos se listan a continuación:

- Permiten realizar control de calidad de alta precisión.
- Reducen el tamaño de las líneas de producción.
- Realizan clasificación automática de los productos gracias a la inteligencia artificial y el machine learning.
- Realizan movimientos a gran velocidad.

La solución que ofrece la empresa Omron para este robot es el llamado X-Delta 3+1, figura 2.15. Omron lo describe como el *sistema de Pick&Place más rápido integrado en la plataforma Sysmac*. Entre sus principales características destacan las siguientes:

- Presenta 3 grados de libertad más un eje rotacional opcional.
- Es capaz de ejecutar hasta 200 ciclos por minuto.
- Tiene un rango de trabajo de 500 a 1.600mm y un rango de carga útil de 1 a 8kg.
- Gracias a la plataforma Sysmac el control de robot está perfectamente integrado en el controlador de máquina NJ y es capaz de controlar hasta 8 robots a la vez.

5. ROBOT SCARA

El robot SCARA viene del acrónimo en inglés "Slective Compliant Assembly o Articulated Robot Arm" que en español significa "Brazo de Robot de Montaje Selectivamente Adaptable". Estos robots destacan por tener pequeñas dimensiones y ser ideales para llevar a cabo trabajos repetitivos a gran velocidad. Son mundialmente utilizados por las empresas para aplicaciones de coger y dejar piezas de un punto A a un punto B (Pick&Place). El brazo robótico le permite automatizar procesos repetitivos mejorando la eficiencia y la productividad de las líneas. Además, el robot SCARA presenta grandes ventajas frente a los robots cartesianos de 3 ejes, usados tradicionalmente para la manipulación de piezas, gracias a sus cuatro grados de libertad que ofrecen mayor rapidez, precisión y movimiento [8].

Están compuestos por dos brazos mecánicos unidos a una base integrada en el propio robot. Para poder controlar el movimiento horizontal del robot tienen incorporados motores que utilizan la cinemática

inversa y la interpolación en los ejes 1 y 2. Además, el eje Z dispone de un movimiento rotativo sobre su propio eje, figura 2.17.

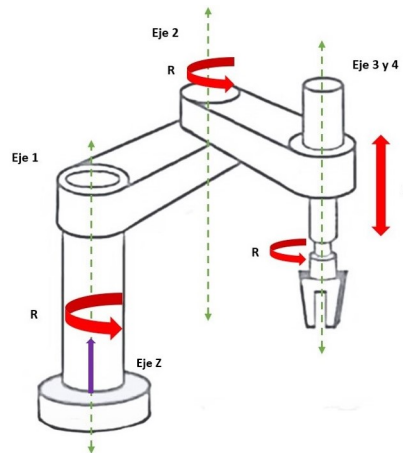


Figura 2.16: Ejes de rotación y desplazamiento del robot. (Fuente:[8])

Dependiendo de las características del producto que fluye por la cinta, el robot SCARA puede presentar en la parte final del brazo una pinza o chupón de succión para recoger los productos.



Figura 2.17: Robot SCARA Serie X. (Fuente:[6])

Omron dispone de una amplia gama de Robots Scara de 4 ejes. En concreto, el que se observa en la figura 2.17 pertenece a la serie X, presenta una mayor rigidez, velocidad y precisión frente a otros modelos de la misma compañía. Además, posee funciones integradas de visión para hacer el seguimiento de los productos por la cinta transportadora. El robot se puede ajustar según la aplicación para tener un alcance de entre 20mm y 1200mm, y una carga útil de 1 kg a 50 kg [6].

2.4 Sysmac Studio

Sysmac es una plataforma para la automatización que proporciona control y gestión integral de una planta de automatización, como la vista en el apartado anterior, figura 2.10.

El concepto de *Sysmac* se basa en tres pilares básicos:

- Un único controlador de máquina que integra lógica, motion, seguridad y control síncrono para toda la plataforma, figura 2.12.

- Un único Software, Sysmac Studio, que proporciona un entorno de desarrollo integrado para configuración, programación y monitorización, que se desarrollará más adelante con mayor detalle.
- Una única conexión de tres redes IO-Link, EtherCAT y Ethernet/IP. Figura 2.18.



Figura 2.18: Solución completa: Plataforma de automatización integrada. (Fuente:[6])

En este apartado se va a explicar de manera detallada el software Sysmac Studio ya que es la herramienta que se ha utilizado para la aplicación de visión artificial y de Pick&Place de este proyecto.

Omron describe su software Sysmac Studio como un entorno de desarrollo integrado (IDE) que integra lógica, motion, robótica, HMI, visión, detección, seguridad y simulación en 3D en una única plataforma.

Sysmac Studio cumple abiertamente con el estándar IEC 61131-3 que tiene ventajas como una mayor legibilidad, reutilización de códigos, variedad de lenguajes de programación, portabilidad de programas de distintos vendedores, etc.

El modelo de Software que se especifica en IEC 61131-3 y que por tanto cumple Sysmac Studio se muestra en la figura 2.19.

Tanto los programas, las funciones (FUN) y los bloques de funciones (FB) son llamados **POU (Program Organization Unit)**. En los programas es donde se encuentra la secuencia lógica y el motion, y para poder especificar el tiempo de ejecución debe estar asignado a una tarea. En otras palabras, es la tarea, y no el programa, la que se ejecuta. A una única tarea pueden asignarse hasta 128 programas diferentes. Además, los programas siempre contienen una tabla de variables locales y un algoritmo donde se puede utilizar cualquier función (FUN) o bloque de función (FB), como se observa en la figura 2.20.

Sabiendo lo que es un programa falta nombrar algunas de las diferencias que existen entre una función y un bloque de función para entender en el próximo capítulo la programación de la aplicación:

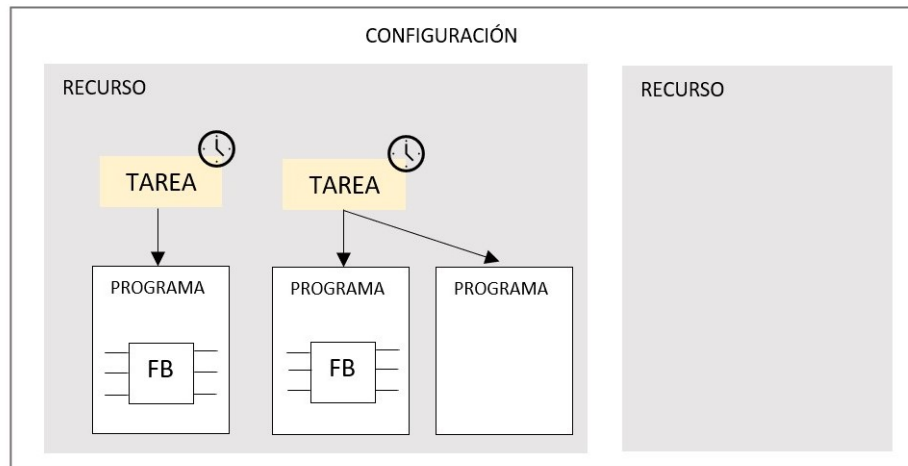


Figura 2.19: Modelo Software IEC 61131-3.

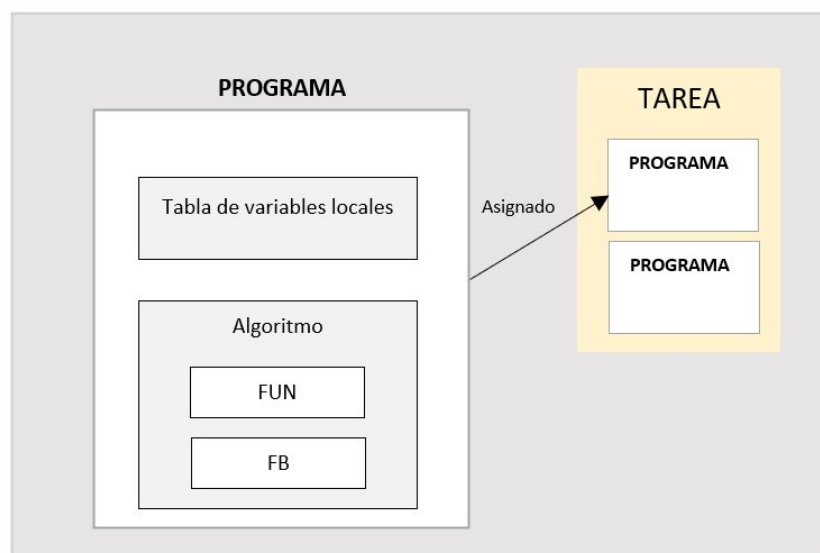


Figura 2.20: Esquema de un programa asignado a una tarea.

Función (FUN)

- Es más fácil de introducir ya que no se necesita instancias.
- Pueden usarse de forma ilimitada ya que no ocupan memoria.
- Es ideal para utilizar en cálculos simples e instrucciones que no requieran memorizar el estado.
- No permite utilizar dentro de su algoritmo bloques de funciones.

Bloque de función (FB)

- Necesita instancias para poder usarlas.
- No pueden usarse de manera ilimitada porque, al contrario de las funciones, si ocupan memoria.

- Se utilizan para instrucciones donde se requiera memorizar estados internos, como temporizadores.
- Permite utilizar dentro de su algoritmo tanto funciones como bloques de funciones.

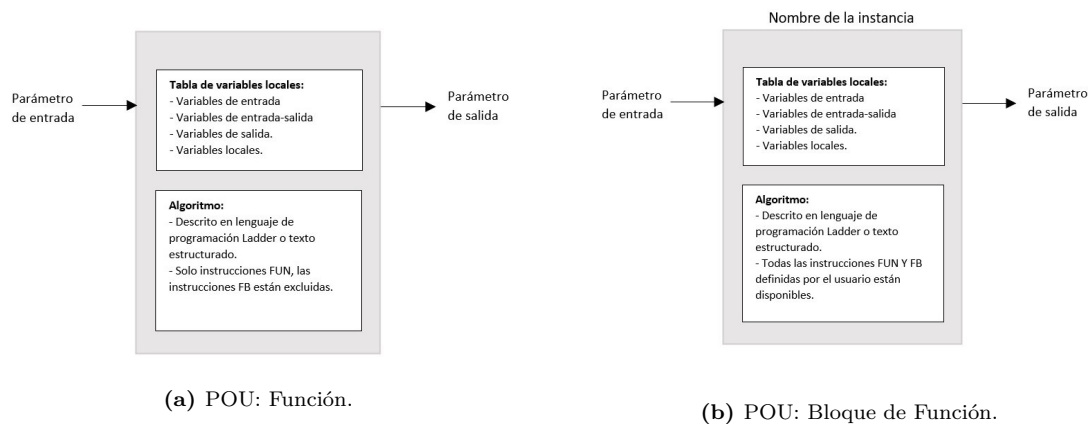


Figura 2.21: Estructuras general de una función (FUN) y de un bloque de función (FB)

En las figuras 2.21a Y 2.21b se pueden observar la estructuras, de forma general, de una función y de un bloque de función. Para esta aplicación, que se desarrolla en el siguiente capítulo, únicamente se usarán bloques de funciones que ya están preconfigurados para nuestra aplicación en concreto y cuyos detalles internos son una caja negra para el usuario.

El algoritmo en el POU puede expresarse hasta con cinco lenguajes de programación diferentes según el estándar IEC 61131-3:

- LD (Diagrama ladder)
- ST (Texto estructurado)
- SFC (Diagrama de funciones secuenciales)
- FBD (Diagrama de bloques de funciones)
- IL (Listas de instrucciones)

Sin embargo, los únicos lenguajes de programación que acepta el controlador de máquina NJ, figura 2.12, son el **lenguaje de diagrama ladder (LD)** y el **texto estructurado (ST)**. Además, Sysmac Studio permite mezclar ambos lenguajes dentro de un mismo algoritmo. En las figuras 2.22 y 2.23 se puede ver un ejemplo de ambos lenguajes.

En cuanto a las variables, en Sysmac Studio, se definen siempre por su nombre, tipo y atributos. Como en todo entorno de programación las variables son imprescindibles para intercambiar información (E/S) con dispositivos externos, efectuar operaciones de cálculo...

Existen tres tipos de variables que podemos encontrar en el sistema:

- **Variables definidas por el usuario:** El programador es el que define el nombre, el tipo de datos y los atributos.

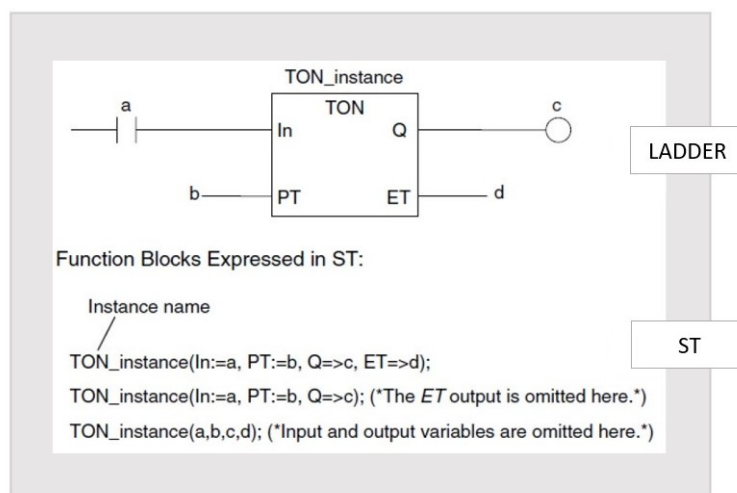


Figura 2.22: Bloque de función «TON» en lenguaje *Ladder* y *ST*.

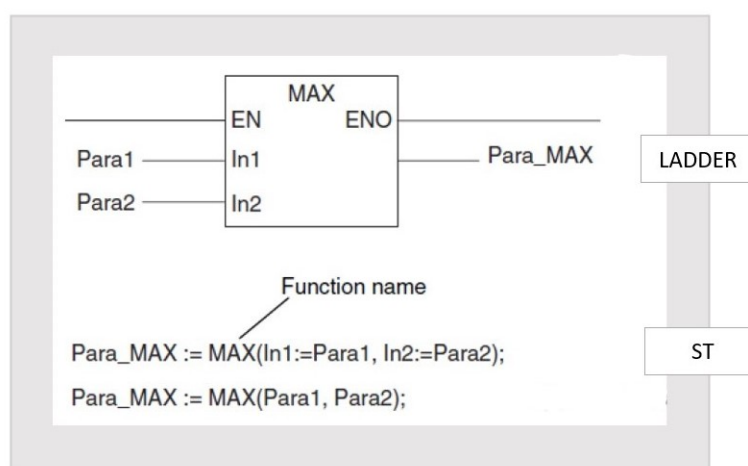


Figura 2.23: Función «MAX» en lenguaje *Ladder* y *ST*.

- **Variables semi-definidas por el usuario:** Se utilizan principalmente para acceder a dispositivos específicos. No se les puede asignar cualquier atributo, solo algunos están disponibles.
- **Variables del sistema:** Los nombres y los atributos de estas variables ya vienen definidos para funciones específicas.

Dentro de las variables definidas por el usuario se puede hacer una clasificación en función de en qué parte del sistema se puedan utilizar:

- **Variable Local:** Este tipo de variable únicamente se puede leer o escribir desde la misma POU donde están definidas.
- **Variable Global:** Es accesible desde cualquier lugar del sistema.
- **Variable Externa:** Estas variables se utilizan para poder acceder a datos que están fuera de la POU, como a las variables globales. Hace de nexo de unión entre variables locales y variables globales como se ve en la figura 2.24.

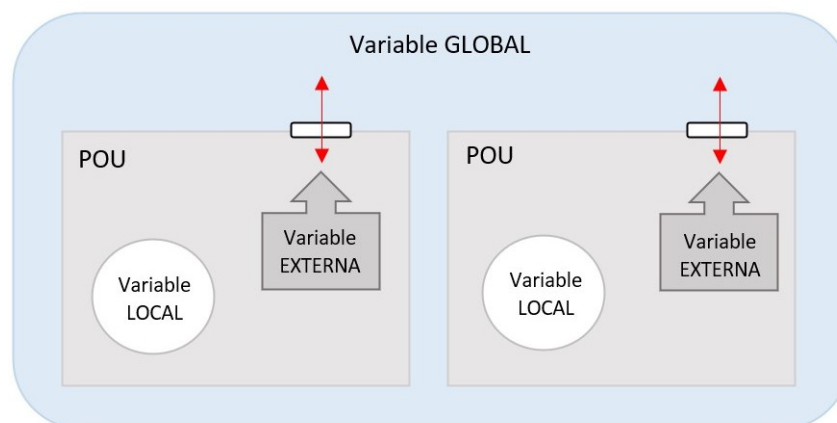


Figura 2.24: Tipo de variables definidas por el usuario.

Entorno de simulación

En el momento en el que se accede al entorno de simulación y se escoge la opción de *crear un proyecto nuevo* se debe elegir el controlador de máquina con el que se va a trabajar.

En la primera pantalla emergente se pueden distinguir varios elementos que se muestran en la figura 2.25, cuyas características se irán viendo a lo largo de este apartado.



Figura 2.25: Ventana principal Software Sysmac Studio.

El primer paso, siempre que se está trabajando con un nuevo proyecto, es introducir los dispositivos que, junto al controlador, se necesitan para el correcto funcionamiento de la aplicación. Esta configuración se realiza pulsando sobre la pestaña **Bastidores de expansión CPU**. Por defecto, aparece el controlador que se ha seleccionado previamente, figura 2.26, y a partir de ahí se le pueden añadir diferentes dispositivos, como módulos de entradas y salida, figura 2.27.

Si, por el contrario, la aplicación requiere conectarse a una máquina real, como la que se analizó en el apartado anterior, figura 2.10, se deberá escoger la opción de la barra de herramientas **Controlador** → **Configuración de comunicaciones** y, a través del cable EtherCAT y la opción **Transferir** se realizará la conexión entre el software y la máquina, cargando todos los elementos en el sistema, figura 2.28.

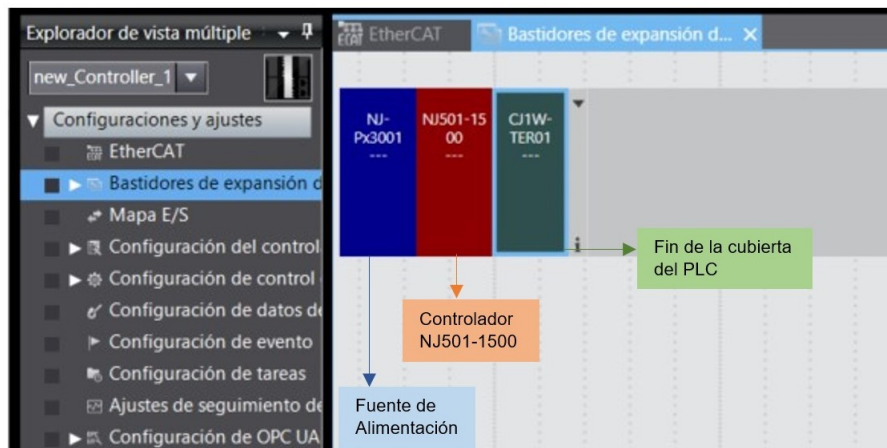


Figura 2.26: Bastidor CPU.

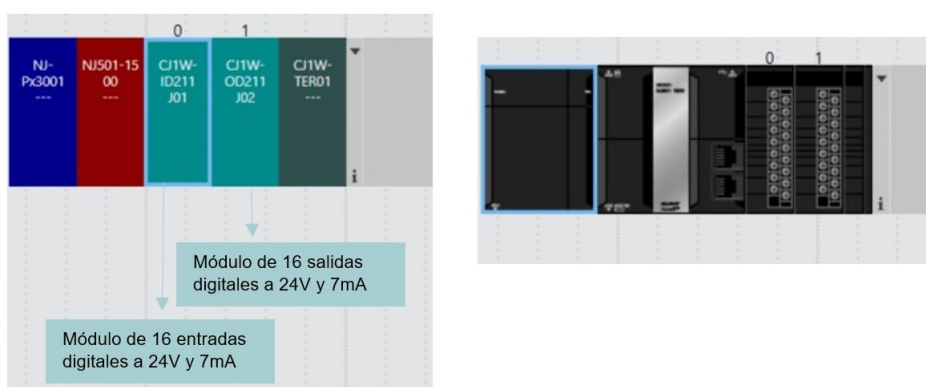


Figura 2.27: Módulo de entradas y salidas añadido al controlador.

Tanto si se han añadido los dispositivos uno a uno o se han transferido desde la máquina real, en el segundo paso se deben comprobar las variables de entrada y salida de cada dispositivo pulsando en la pestaña **Mapa E/S**. Las variables que aparecen en el Mapa E/S son asignadas directamente por el programa a un estado de señal correspondiente a una salida o una entrada, sin embargo, en el entorno de programación se podrán definir otro tipo de variables para almacenar resultados.

Para la creación de un programa hay que desplegar la opción de **POUs** y seleccionar **Añadir programa** haciendo pulsando sobre el botón derecho del ratón. Una vez creado el programa, aparece una pantalla en blanco para introducir el algoritmo de programación. Como se ha visto anteriormente se pueden utilizar dos tipos de lenguajes: Ladder y texto estructurado. Lo más sencillo para que el código sea fácilmente legible es utilizar ladder, o un conjunto entre ladder y texto estructuro. En la ventana de programación se pueden crear funciones y bloques de funciones propias, aunque, para la aplicación que se va a llevar a cabo, se utilizan bloques de funciones de robótica creados previamente por Sysmac.

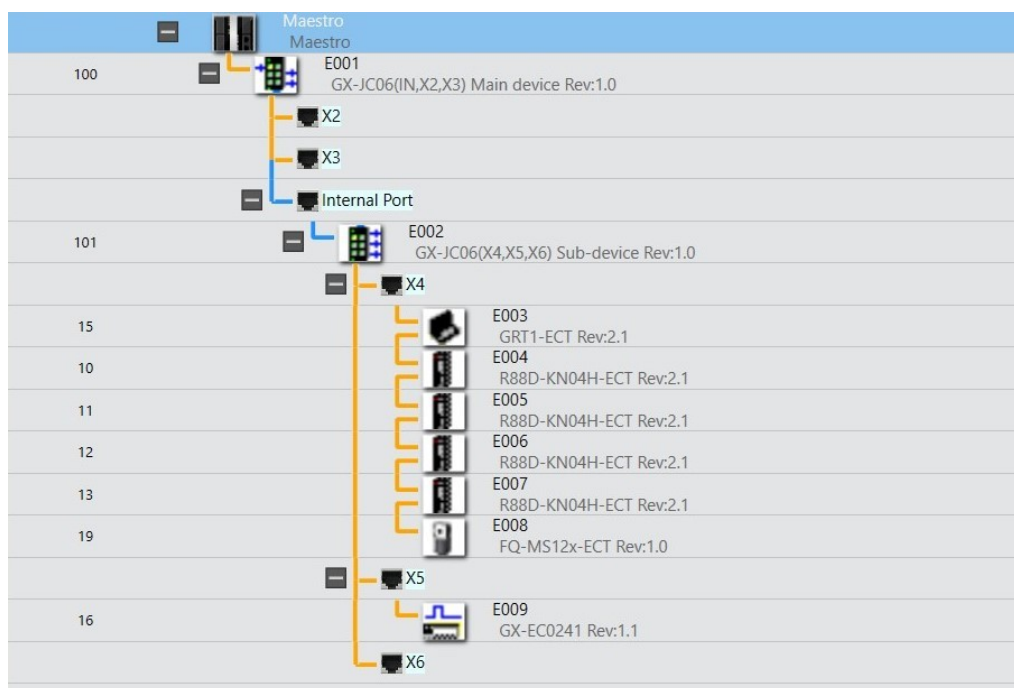


Figura 2.28: Red EtherCAT con todos los dispositivos de la máquina real.

Para introducir funciones o bloques de funciones se pulsa el botón derecho del ratón sobre la ventana de programación y se selecciona el elemento que se quiere introducir, figura 2.29.

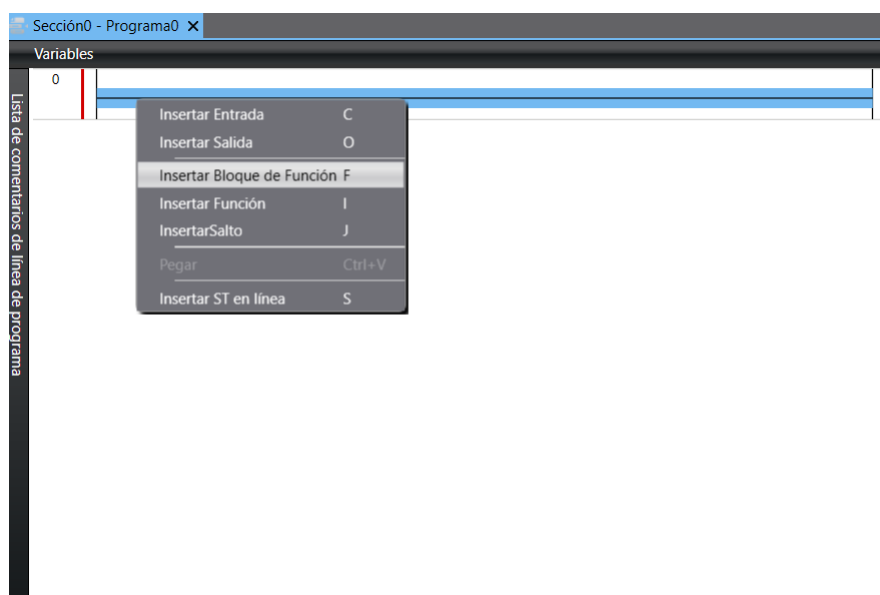


Figura 2.29: Insertar una FUN o un FB en el programa.

Una vez que el programa está terminado se debe asignar a una tarea para que pueda ser ejecutado. Si solo existe un único programa en el proyecto automáticamente se asignará a la tarea principal.

En el capítulo 4 se extenderá la información del software Sysmac Studio ya que se detallará de forma exhaustiva cómo se ha realizado la parte práctica de este proyecto.

Capítulo 3

Desarrollo

En este apartado se explica de manera detallada el objetivo práctico de este proyecto y los pasos necesarios para conseguirlo. En el capítulo 4 se verán los resultados que se han obtenido.

3.1 Aplicación de Trazabilidad

Como se ha comentado en otras secciones de este trabajo, para la aplicación de trazabilidad que se va a llevar a cabo se han utilizado tanto el sensor de visión FQ-M, figura 2.13, como la plataforma de software Sysmac Studio, figura 2.25, ambas soluciones para la automatización de líneas de producción de la empresa Omron. El principal objetivo de este apartado, es, por lo tanto, entender en profundidad el empleo de ambos sistemas de una forma integrada.

El sensor de visión, a partir de métodos de detección que se describen en el apartado 3.1.3, detecta piezas en la cinta que presentan una similitud a un modelo configurado previamente a través de Sysmac Studio. La posición de las piezas detectadas son mandadas al controlador o robot para que las retire de la línea de producción, realizando el movimiento de Pick&Place.

Para la realización de esta aplicación se han seguido los pasos enumerados a continuación:

1. Conexión de software y máquina.
2. Calibración.
3. Métodos de detección.
4. Transmisión de datos de salida.

3.1.1 Conexión de software y máquina

El primer paso, para poder conectar Sysmac Studio con el sensor de visión y poder trabajar con él, es conectarse a través de EtherCAT con toda la máquina y transferir todos los elementos de la máquina al simulador.

Pulsando en **Controlador** → **Configuración de conexión** se debe seleccionar la opción **Conexión Ethernet a través de un hub** y poner la dirección IP que utilice la máquina, figura 3.1.

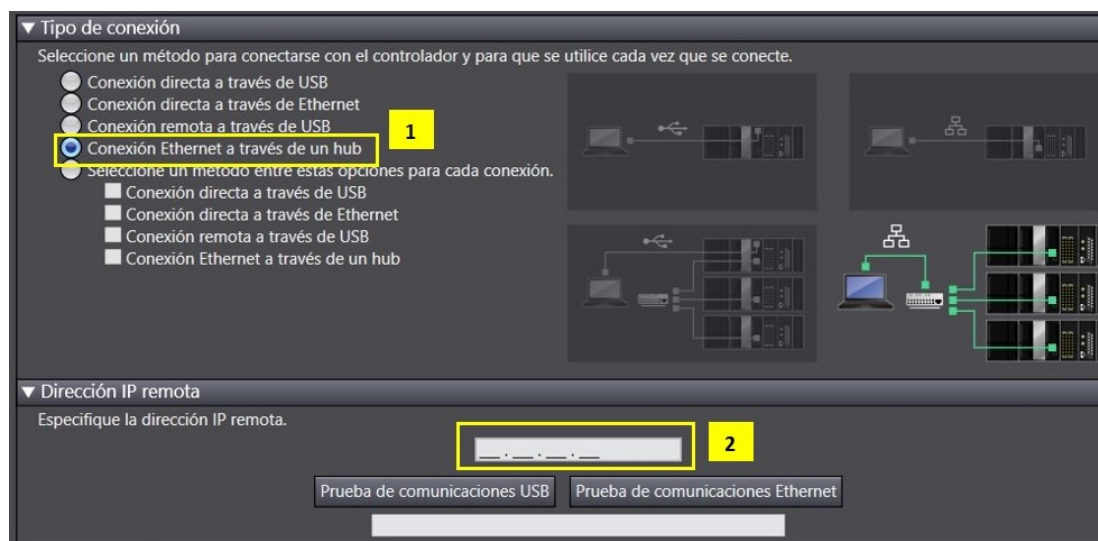


Figura 3.1: Conexión Ethernet software-máquina.

Para completar la conexión se pulsa en el botón **online** y, si ha detectado un controlador al que conectarse, se pulsa en **Controlador** → **Transferir** → **Desde controlador** para que todos los dispositivos conectados al controlador aparezca en el simulador y se pueda trabajar con ellos.

De todos los dispositivos a los que se conecta solo nos interesa el sensor de visión FQ-M, figura 3.2.

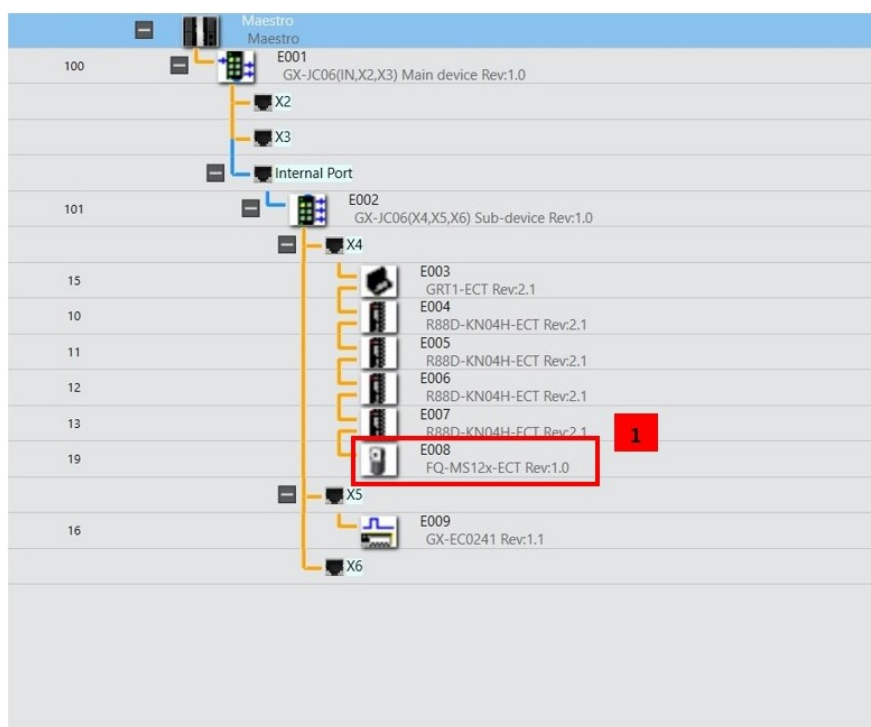


Figura 3.2: Red EtherCAT con conexión al sensor FQ-M de la máquina real.

Para acceder a la pantalla de trabajo se debe identificar el módulo de FQ-M en la ventana de **Configu-**

ración y ajustes, y seleccionar la **escena cero** para empezar a trabajar, como se muestra en el *punto 1* de la figura 3.3.

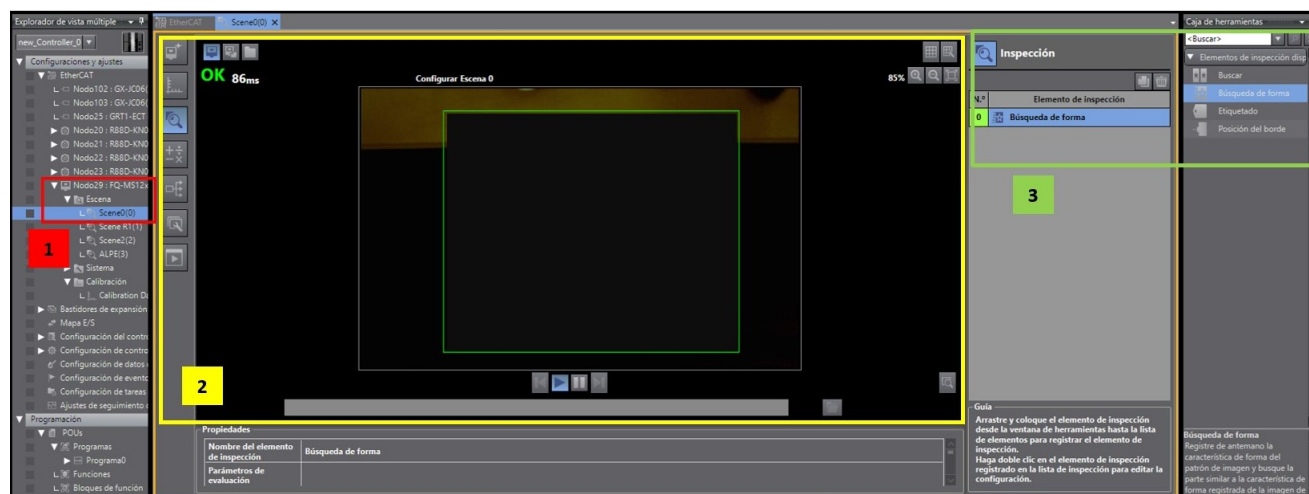


Figura 3.3: Pantalla de trabajo FQ-M.

En la figura 3.3 también se muestra el *punto 2*, **Panel de edición**, que se utiliza para editar y ver los datos seleccionados en el panel del explorador. El panel de edición consta principalmente de imágenes, configuraciones, propiedades y orientación. El *punto 3* es la **Caja de herramientas** donde se muestra una lista de todos los elementos de inspección que se pueden agregar a la escena.

3.1.2 Calibración

La calibración se utiliza para convertir las coordenadas de la cámara en coordenadas reales. El sensor de visión es compatible con dos tipos de calibración:

- **Calibración de propósito general:** Este tipo de calibración convierte las coordenadas de la cámara del sensor de visión en coordenadas reales con el mismo origen. Por lo tanto, genera la posición detectada en coordenadas reales.
- **Calibración para *Conveyor tracking*:** Este tipo de calibración utiliza un codificador para convertir las coordenadas de la cámara del sensor de visión en coordenadas para el robot colocado en la misma cinta. Por lo tanto, genera la posición detectada en coordenadas del robot.

Para nuestra aplicación se usará la calibración por ***Conveyor tracking*** ya que resulta más interesante para posibles aplicaciones futuras. Existen dos maneras de realizar este tipo de calibración:

- **Muestreo por lote:** Utiliza una plantilla específica para la calibración.
- **Especificación de puntos:** sin utilizar plantilla de calibración se establecen los puntos que el usuario desee.

Para este caso se utilizará la calibración por **Muestreo por lote** cuyo flujo de trabajo a seguir para realizar correctamente la calibración se muestra en la figura 3.4.

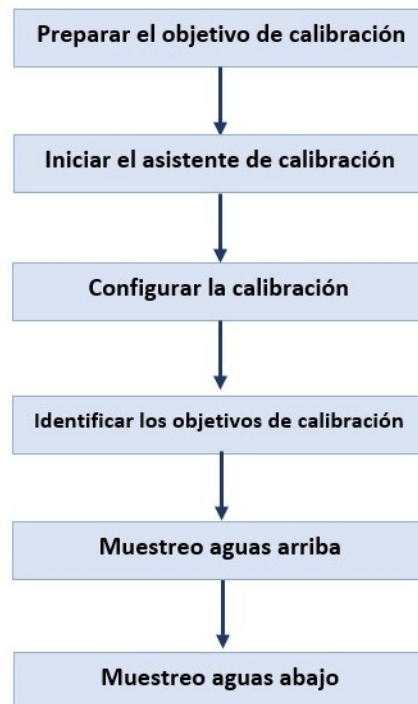


Figura 3.4: Flujo de trabajo para realizar la calibración *Muestro por lote*.

El manual de Sysmac Studio advierte que antes de realizar cualquier calibración es necesario establecer el máximo valor al contador del encoder, *1.000.000.000*, para que la calibración se realice correctamente. Una vez acabado el proceso se debe retornar el valor del encoder a su valor original.

PASO 1: PREPARAR EL OBJETO DE CALIBRACIÓN

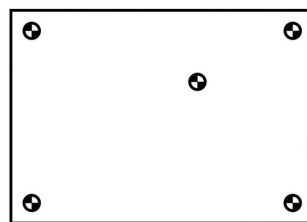


Figura 3.5: Plantilla para la calibración.

Se imprime el objetivo de calibración (plantilla) y se coloca en el campo de visión de la cámara. La plantilla se descarga de la página oficial de Omron y se pueden elegir las dimensiones según el campo de visión de la cámara a utilizar.

PASO 2: INICIAR EL ASISTENTE DE CALIBRACIÓN

Se selecciona dentro de **configuración y ajuste** → **FQ-M** → **Calibración** → **Calibración Data 2**, figura 3.6.

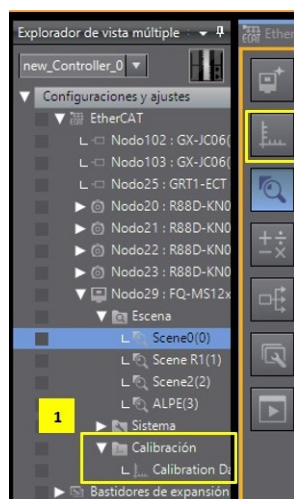


Figura 3.6: Ruta para acceder al asistente de calibración.

PASO 3: CONFIGURAR LA CALIBRACIÓN

Se le da un nombre a la calibración y se selecciona el tipo de calibración que se va a llevar a cabo. Para esta aplicación *Conveyor tracking* → **Muestreo por lote**, figura 3.7.

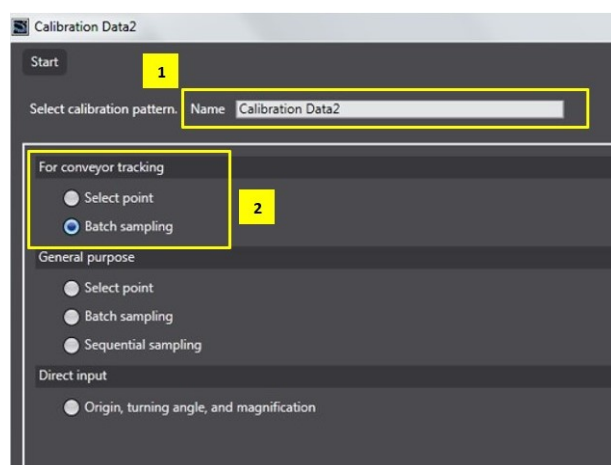


Figura 3.7: Configuración de tipo de calibración.

Se pulsa el botón de siguiente y aparece la pantalla de la figura 3.8. Una vez en la nueva pantalla se debe seleccionar el número de puntos de muestreo, en este caso cuatro para ser más precisos. Y, por último, se especifica el método de entrada para las coordenadas del robot. Para este caso se selecciona **entrada por teclado** para introducir los valores directamente desde el teclado.

PASO 4: IDENTIFICAR LOS OBJETIVOS DE CALIBRACIÓN

Se ajusta la posición del objeto de calibración de modo que las cinco marcas sean visibles por el sensor y se pulsa siguiente para que las marcas sean detectadas, figura 3.9.

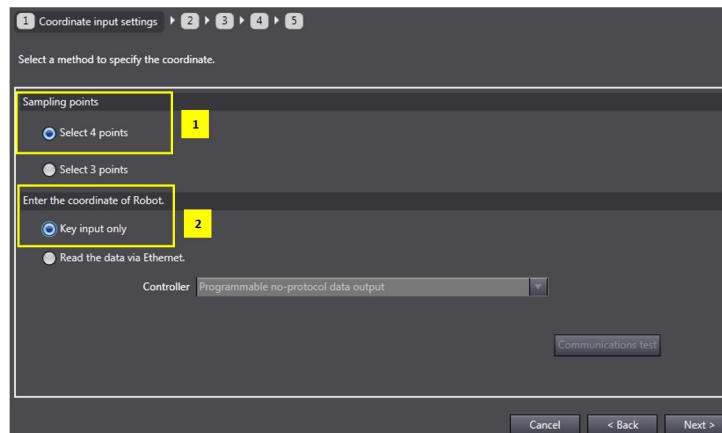
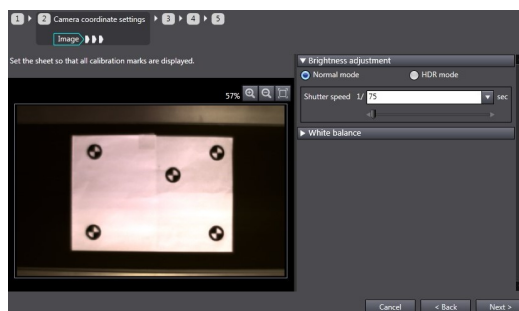
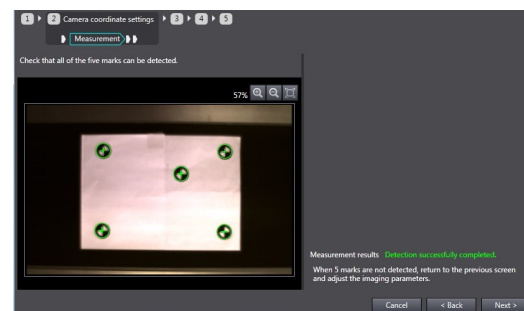


Figura 3.8: Configuración número de puntos de muestreo.



(a) Ajuste del objeto de calibración al campo de visión.



(b) Detección del objeto de calibración.

Figura 3.9: Identificación del objeto de calibración (plantilla).

PASO 5: MUESTRO AGUAS ARRIBA

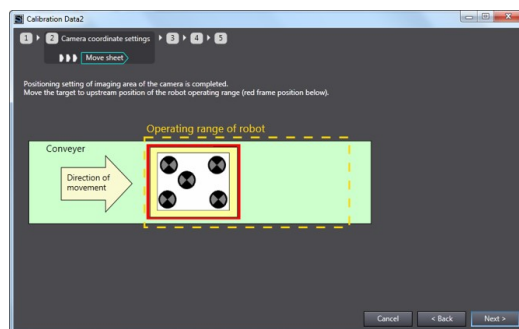
Se mueve la cinta transportadora con el objeto de calibración hasta el inicio de movimiento del robot por la cinta, figura 3.10a. A continuación, se desplaza el centro de la mano del robot a la primera marca de la plantilla y se apuntan las coordenadas X e Y del robot en el campo **coordenadas del robot** (para leer las coordenadas del robot hay que pulsar el botón **leer robot**), figura 3.11.

Se repite el mismo proceso para el resto de puntos de forma ordenada, según la figura 3.10b, apuntando las coordenadas del robot en cada posición.

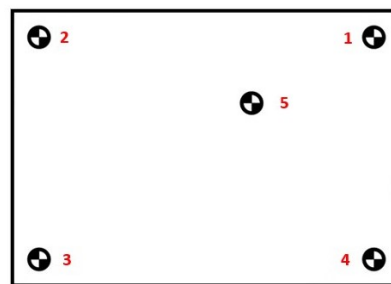
PASO 6: MUESTRO AGUAS ABAJO

Se mueve la cinta transportadora con el objeto de calibración hasta el final de movimiento del robot por la cinta, figura 3.12a. A continuación, se desplaza el centro de la mano del robot a la primera marca del objeto de calibración y se apuntan las coordenadas X e Y del robot en el campo **coordenadas del robot** (para leer las coordenadas del robot hay que pulsar el botón **leer robot**). Se repite el mismo proceso para el resto de puntos de forma ordenada, según la figura 3.10b, apuntando las coordenadas del robot en cada posición.

Una vez realizada la calibración, ambos, sensor y robot, están alineados.



(a) Movimiento del objeto de calibración al principio del alcance del robot.



(b) Numeración de los puntos a detectar del objeto de calibración.

Figura 3.10: Muestreo aguas arriba.

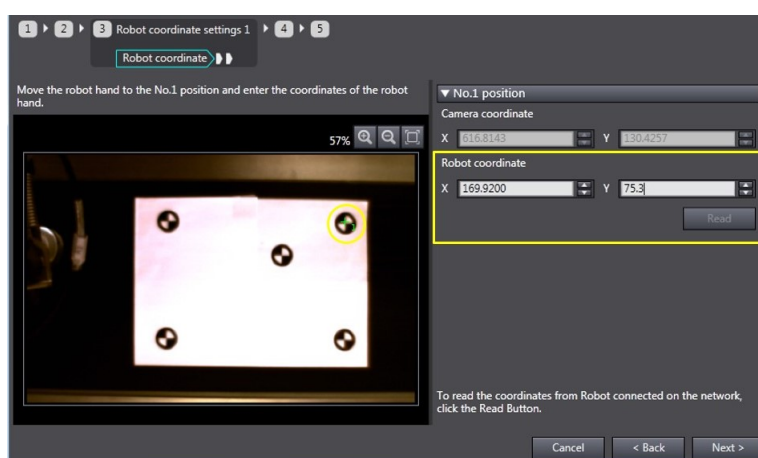
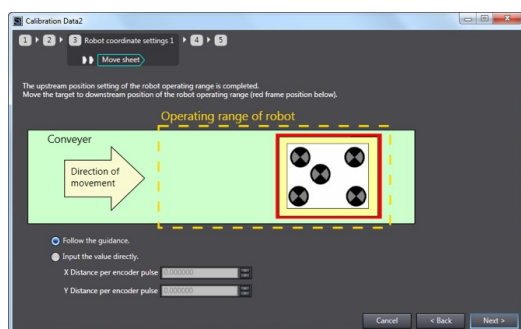
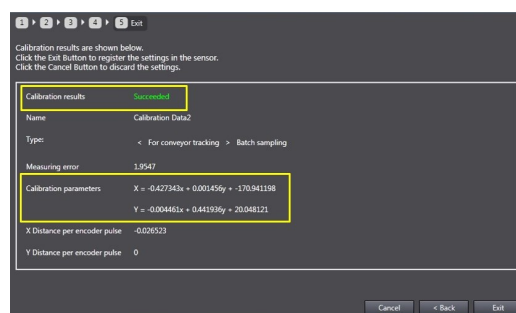


Figura 3.11: Anotación de las coordenadas del robot



(a) Movimiento del objeto de calibración al final del alcance del robot.



(b) Ventana de comprobación de la calibración

Figura 3.12: Muestreo aguas abajo.

3.1.3 Métodos de detección

Existen cuatro métodos de inspección diferentes, figura 3.13, que se utilizan en función de las características del objeto a detectar. Se entiende como método de inspección a la forma en la que se configura el modelo que, más tarde, el sensor de visión debe comparar con los productos que circulan por la línea. Si el producto es igual que el modelo entonces el juicio se dice que es **OK**, sin embargo, si el producto no

cumple con las características del modelo se dice entonces que el juicio es **NG**.



Inspección	Ejemplo	Elemento de inspección utilizado
Detección de posiciones con formas a alta velocidad	Los objetos de medición de la misma forma pueden ser detectados. 	Búsqueda de formas
Detección de posiciones con patrones	Los objetos de medición del mismo color y patrón pueden ser detectados. 	Búsqueda
Detección de posiciones por bordes	Las posiciones de los bordes de las superficies de cristal pueden ser detectadas. 	Posición por bordes
Detección de posiciones por grupos	Se pueden detectar grupos del mismo color. 	Etiquetado

Figura 3.13: Resumen elementos de inspección.

BÚSQUEDA DE FORMA

En el método de detección por **búsqueda de forma** se guarda como patrón tanto el contorno como el color del modelo. El sensor de visión lo busca en la región de medida que se haya elegido previamente. Tanto en este método, como en el siguiente, **detección de búsqueda** se puede escoger el grado de correlación que se desea que tenga el modelo con el producto detectado. Si el grado de correlación es alto solo se detectarían los productos que sean iguales al modelo, pero si, por el contrario, se pone un grado de correlación menor, los productos detectados no serán exactamente iguales al modelo. En el apartado 4.1 se lleva a cabo un ejemplo real explicando detalladamente los pasos a seguir.

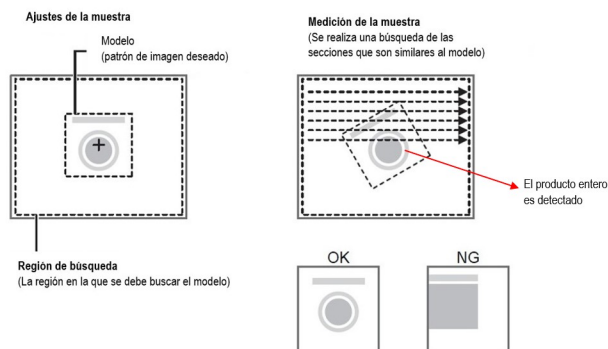


Figura 3.14: Método de detección: Búsqueda de forma.

BÚSQUEDA

En el método de detección por **búsqueda** se guarda como patrón únicamente el contorno del modelo sin fijarse en otras características como el color. Una vez el modelo es establecido se le indica al sensor la región de búsqueda: espacio en el que debe detectar las piezas iguales al modelo. Aunque las piezas, iguales al modelo, se encuentren rotadas pueden ser detectadas por el sensor de visión que, más tarde, manda sus posiciones X e Y y el ángulo de rotación al robot. Es el método más rápido y más usado en aplicaciones de Pick&Place. Además, es muy estable incluso con objetos superpuestos, parcialmente completos y con cambios de iluminación.

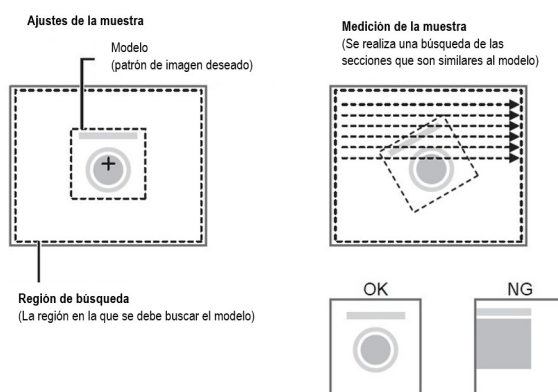


Figura 3.15: Método de detección: Búsqueda.

POSICIÓN POR BORDES

En el siguiente método, **posición por borde**, lo que detecta el sensor es la posición de un borde, que no es más que una zona en la que se produce un gran cambio de color. Por ejemplo, es ideal para detectar si una etiqueta está bien colocada o si una pieza se encuentra bien posicionada, etc. Para que el algoritmo pueda detectar el borde es necesario establecer antes una región de medida y la dirección en la que se vaya a producir el cambio de color.

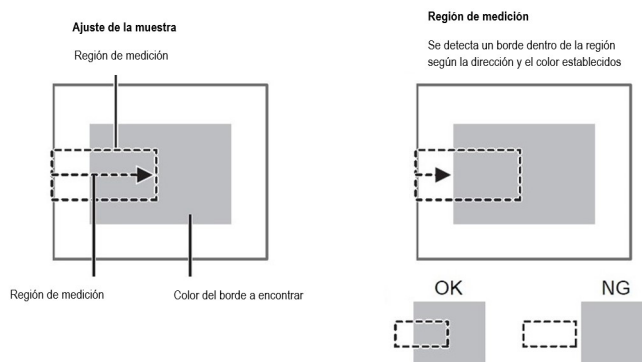


Figura 3.16: Método de detección: Posición por bordes.

ETIQUETADO

Con el método de **etiquetado** se detecta cuantos grupos de color existen dentro de una región de medida, y, como salida, el sensor de visión proporciona la posición de cada uno de esos grupos (*labels*) de forma enumerada. Es necesario establecer el modelo del color que se desea detectar. En el apartado 4.1 se lleva a cabo un ejemplo real explicando detalladamente los pasos a seguir.

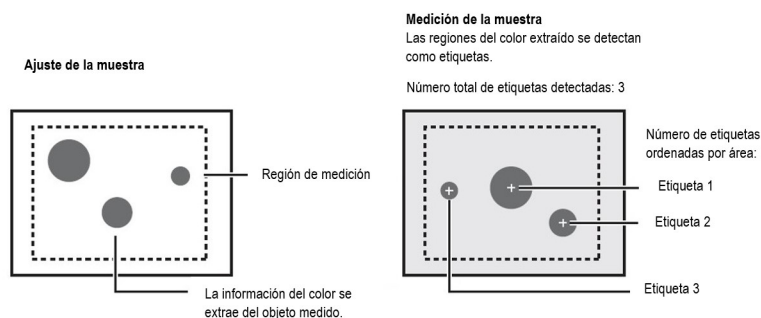


Figura 3.17: Método de detección: Etiquetado.

3.1.4 Transmisión de datos de salida

Los datos de salida del sensor de visión, en su mayoría datos de posiciones de los productos detectados, pueden llegar al controlador o a otros dispositivos a través de cuatro formas distintas:

- Salidas digitales.
- EtherCAT.
- Ethernet PLC Link.
- Ethernet No-Protocol.

En este apartado se describe la comunicación vía EtherCAT, que es uno de los métodos de comunicación más utilizados.

EtherCAT (Ethernet Control Automation Technology) es un sistema de red industrial de alto rendimiento basado en *Sistema Ethernet* y puede realizar comunicaciones más rápidas y eficientes. Cada nodo logra un tiempo de ciclo de comunicaciones corto al transmitir tramas Ethernet a alta velocidad. La eficacia de EtherCAT se puede utilizar plenamente no solo en grandes sistemas de control, que requieren altas velocidades de procesamiento e integridad del sistema, sino también en sistemas de control pequeños [26].

EtherCAT no envía datos a nodos esclavos individuales en la red, sino que pasa tramas Ethernet a través de todos los nodos esclavos. Cuando la trama pasa a través de un nodo esclavo, el nodo esclavo lee y escribe datos en las áreas asignadas a él en unos pocos nanosegundos. Una vez que la trama llega al esclavo final son enviadas de regreso pasando nuevamente por todos los esclavos hasta el maestro EtherCAT. Con esta estructura EtherCAT se asegura la transmisión de datos a alta velocidad y en tiempo real [26].

La comunicación vía EtherCAT entre el maestro EtherCAT y el sensor de visión puede establecerse para controlar operaciones de comunicaciones de comando/respuesta o para enviar datos después de las mediciones. Como se ha visto en el apartado 3.1.1 se ha configurado la comunicación entre el controlador NJ y el sensor de visión con el simulador Sysmac Studio a través de la red EtherCAT.

- **Comunicación comando/respuesta:** Las comunicaciones EtherCAT utilizan objetos de datos de proceso (PDO) para realizar comunicaciones PDO cíclicas. Las señales de control comando/respuesta se manejan almacenando los comandos de control del maestro EtherCAT en el sensor de visión y las respuestas del sensor de visión en las E/S del maestro EtherCAT, como se observa en la figura 3.18, donde el maestro EtherCAT es el controlador NJ.

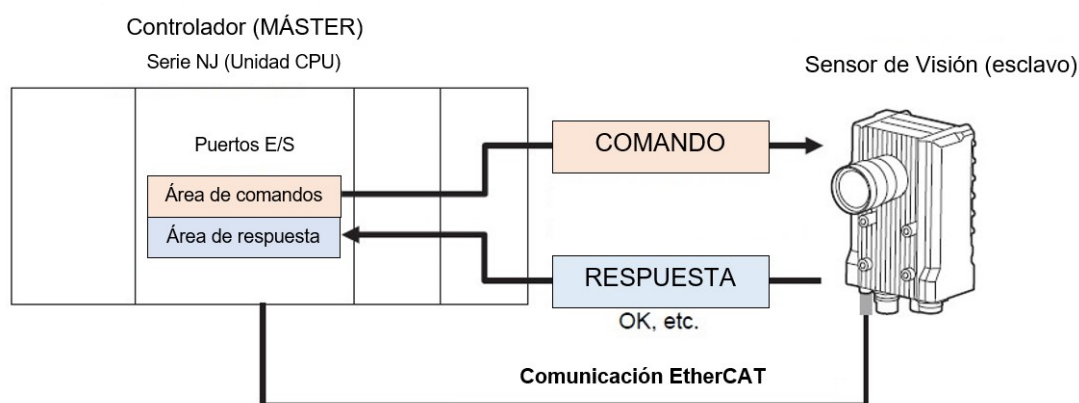


Figura 3.18: Comunicación comando/respuesta: Sensor de visión - Controlador.

- **Salida de datos:** Inmediatamente después de la ejecución de las mediciones, el sensor de visión manda automáticamente los datos de salida que se encuentren en los puertos de E/S del sensor. Esto le permite transferir fácilmente los datos obtenidos de los métodos de inspección al controlador, figura 3.19.

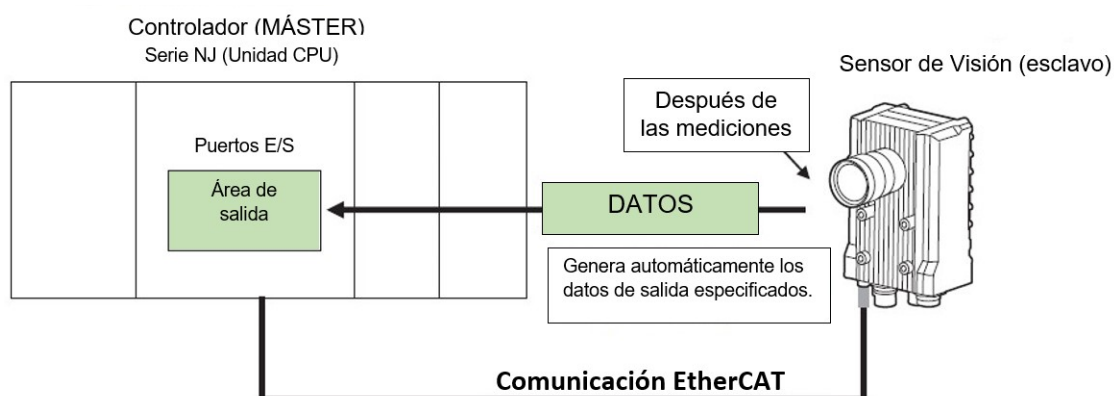


Figura 3.19: Salida de datos: Sensor de visión - Controlador.

Para configurar la red EtherCAT para la comunicación de los datos de salida del sensor se deben seguir los siguientes pasos:

1. Configuración del mapeo EtherCAT PDO en el FQM: Para ello debemos acceder a **Ethercat communication settings** dentro de la escena del sensor de visión, figura 3.20.

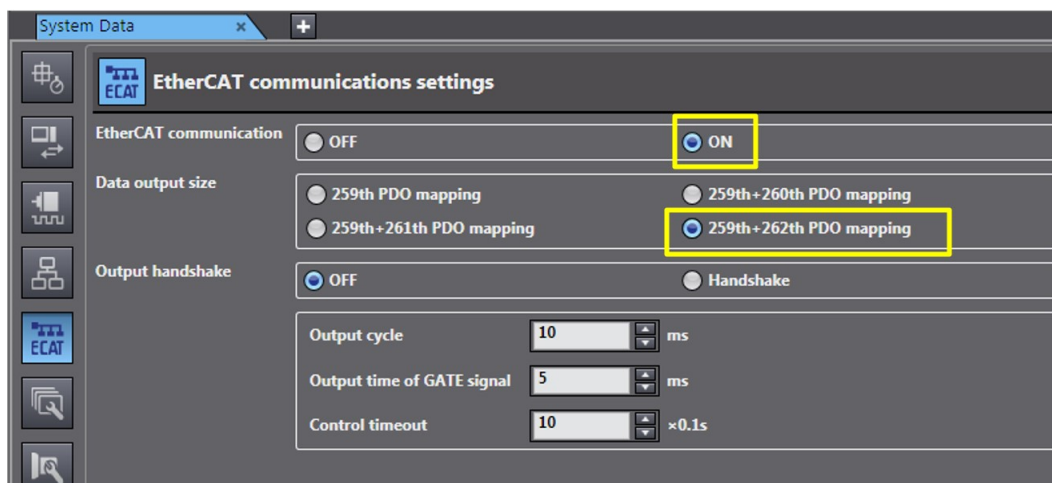


Figura 3.20: Configuración de la comunicación EtherCAT.

2. Configuración del Master EtherCAT según la configuración de PDO: Para ello en la ventana de **EtherCAT** se selecciona el nodo del sensor de visión y se comprueba en la ventana de configuración si el nombre y el nodo son correctos, figura 3.21. A continuación, se pulsa el botón **Editar los ajustes del Mapa PDO**, figura 3.22.

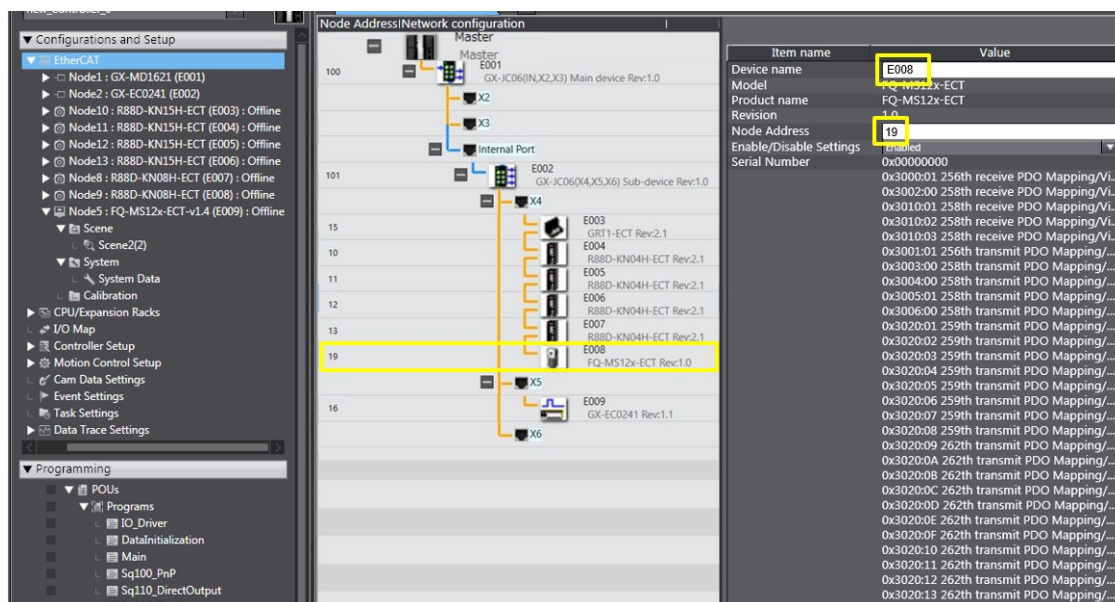


Figura 3.21: Ventana de configuración de la red de dirección del nodo.

3. Definir las variables del dispositivo en el mapa de E/S: En el apartado de **Mapa E/S** se debe comprobar que se muestra el nodo del sensor de visión (en este caso NODO 19) con todas las variables de entrada y salida, figura 3.23 En las variables de salida aparecerán las posiciones de los objetos detectados por el sensor de visión.

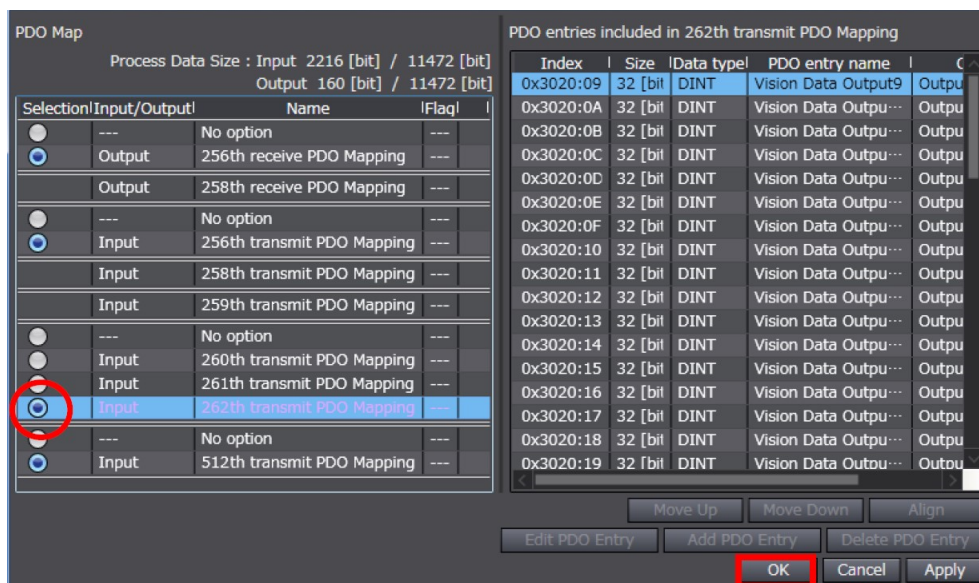


Figura 3.22: Edición de ajustes Mapa PDO.

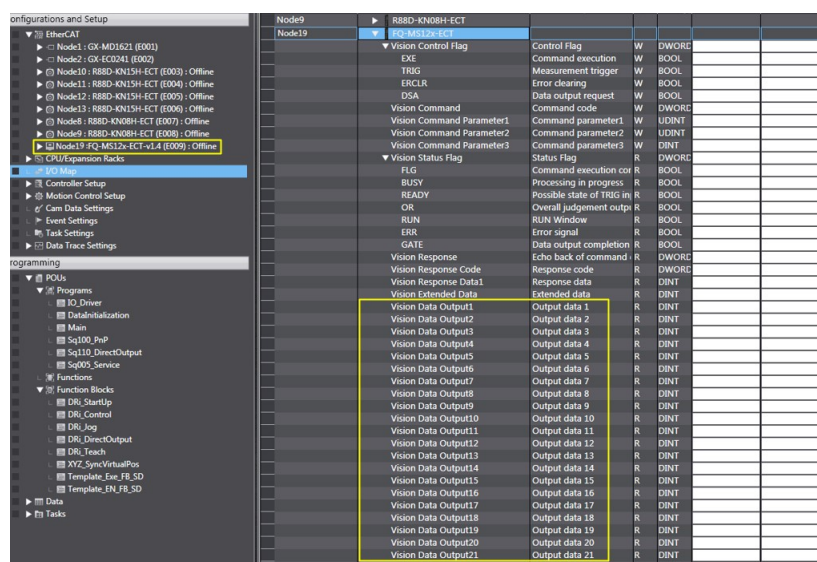


Figura 3.23: Mapa entradas y salidas del Sensor de Visión.

3.2 Aplicación de Pick&Place

En esta sección se va a explicar cómo se ha llevado a cabo el desarrollo de la aplicación de Pick&Place con la utilización del software Sysmac Studio y el paquete Sysmac Studio 3D, ambas herramientas de la empresa Omron.

El simulador Sysmac Studio se explica de forma detallada en la sección 2.4. Omron describe Sysmac Studio 3D como *un entorno de desarrollo integrado (IDE) con simulación en 3D que integra y verifica los movimientos de los robots y los dispositivos periféricos. Puede reproducir en formato digital toda la instalación para verificar el funcionamiento con la misma precisión que una máquina real. Esto ayuda a las empresas a aumentar la productividad incorporando mejoras en la eficiencia de los procesos y en la capacidad de producción.*

Para esta aplicación se va a crear un robot SCARA en 3D que haga la acción de coger y dejar una pieza de un punto A a un punto B. Aunque en el apartado 2.3 se hace una introducción al robot SCARA es necesario explicar más en profundidad el robot y su cinemática para entender el desarrollo de la aplicación.

Un **robot SCARA RPR+R** está formado por una parte fija llamada base y cuatro partes móviles, brazo, antebrazo, Eje Z y Eje R, como se observa en la figura 3.24.

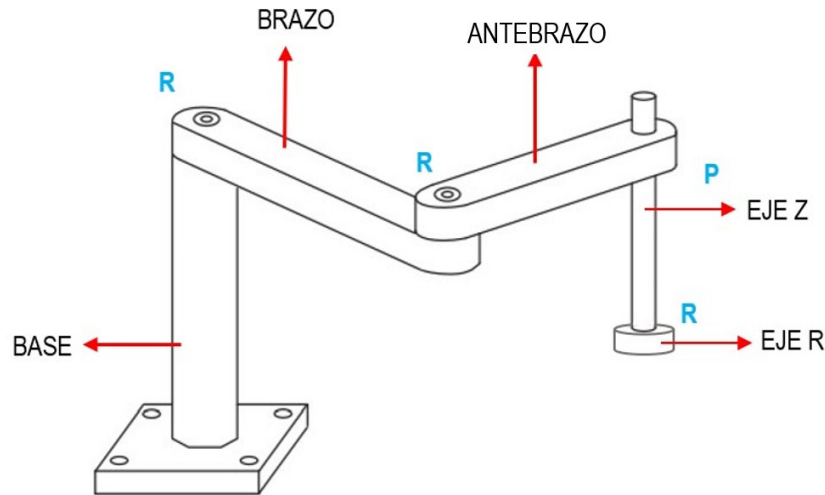


Figura 3.24: Partes ROBOT SCARA.

Se llama robot SCARA RPR+R porque presenta tres movimientos rotacionales y un movimiento periférico en el eje Z, como se observa en la figura 3.25, además, se puede ver tanto el sistema de coordenadas del robot *MCS* cuyo origen siempre se encuentra en la base del robot, como el sistema de coordenadas *ACS* que es un sistema de coordenadas de movimiento lineal o de rotación especificado para cada eje. El controlador de máquina NJ Robotics, figura 2.12, utiliza dichos sistemas de coordenadas para controlar los robots.

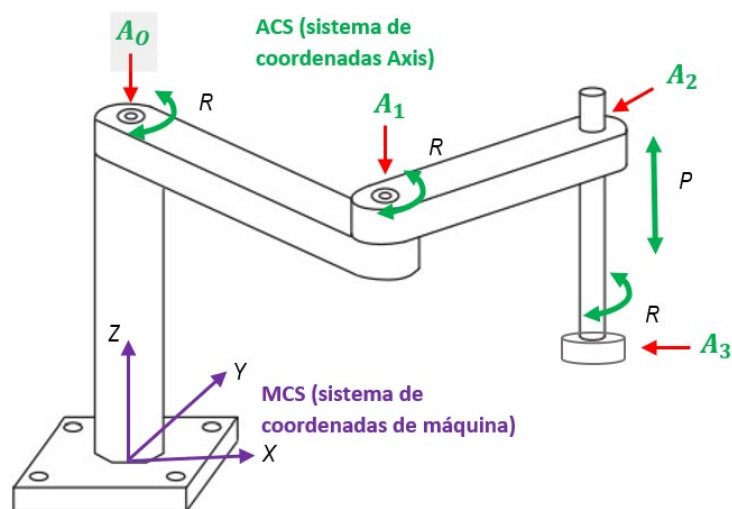


Figura 3.25: Coordenadas ROBOT SCARA.

Para el robot SCARA se define la posición de "0 grados" cuando el enlace $L1$ (brazo) está alineado con el enlace $L2$ (antebrazo). La dirección positiva de cada eje es en sentido antihorario en el plano XY , figura 3.26. La cinemática del ROBOT SCARA RPR+R consta de cuatro ejes y tres parámetros importantes que se definen a continuación:

- **L1**: Longitud del primer enlace.
- **L2**: Longitud del segundo enlace.
- **d1**: Distancia del efector final desde la base (Z) cuando el eje $A2 = 0$.

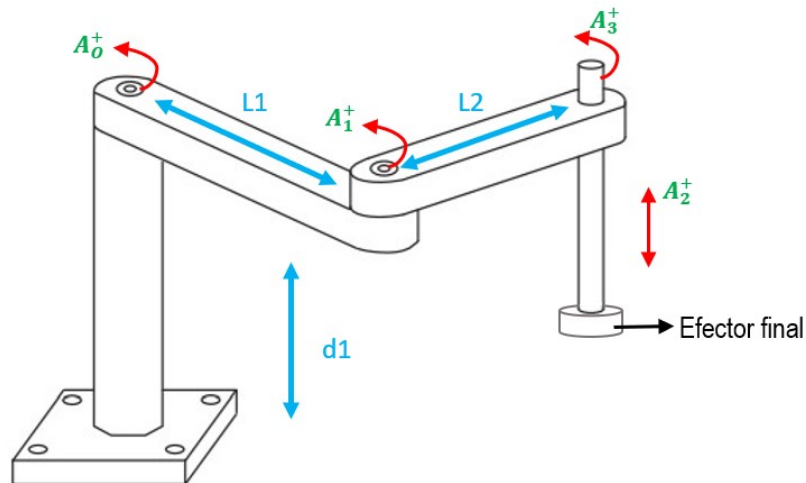


Figura 3.26: Cinemática ROBOT SCARA.

Los pasos a seguir para llevar a cabo el desarrollo de la aplicación se muestran en la figura 3.27, y se desarrollarán uno a uno en los siguientes subapartados.

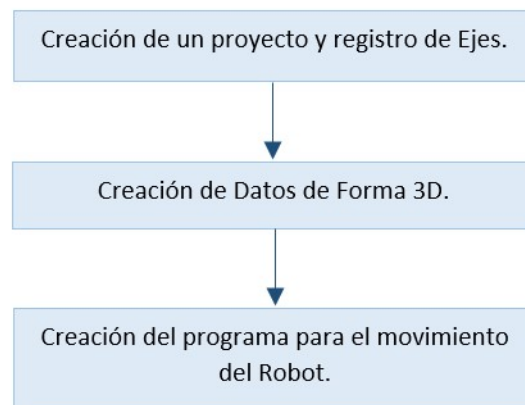


Figura 3.27: Pasos para el desarrollo del ROBOT SCARA.

3.2.1 Creación del proyecto y registro de ejes

El primer paso es crear un proyecto eligiendo el controlador de máquina NJ501-4500, ya que tiene por defecto el módulo de robótica integrado para poder utilizar los bloques de funciones que se describen en

el apartado 3.2.3.

Como se ha visto en la figura 3.25, el robot SCARA está formado por 4 ejes, por lo que se deben crear y configurar cuatro ejes en el sistema. Los ejes del robot operan de forma independiente por lo que cada eje actuará según sus ajustes establecidos.

Para la creación de un eje nuevo se pulsa el botón derecho en **configuración de ejes** → **Añadir** → **Eje de control de movimiento**, figura 3.28.

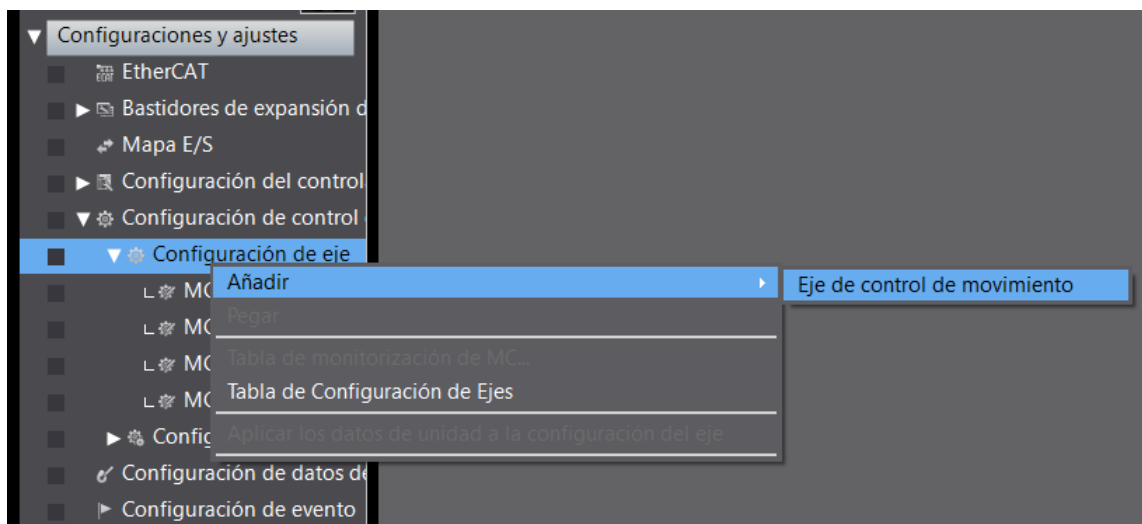


Figura 3.28: Añadir un eje nuevo al sistema.

Una vez creados los cuatro ejes se deben configurar algunos de sus parámetros. En la pantalla de **configuración básica del eje**, figura 3.29, se debe especificar el número de eje, habilitar el eje para ser utilizado y elegir el tipo de eje. En este caso, como se trabajará con una simulación, se utilizará un eje virtual.

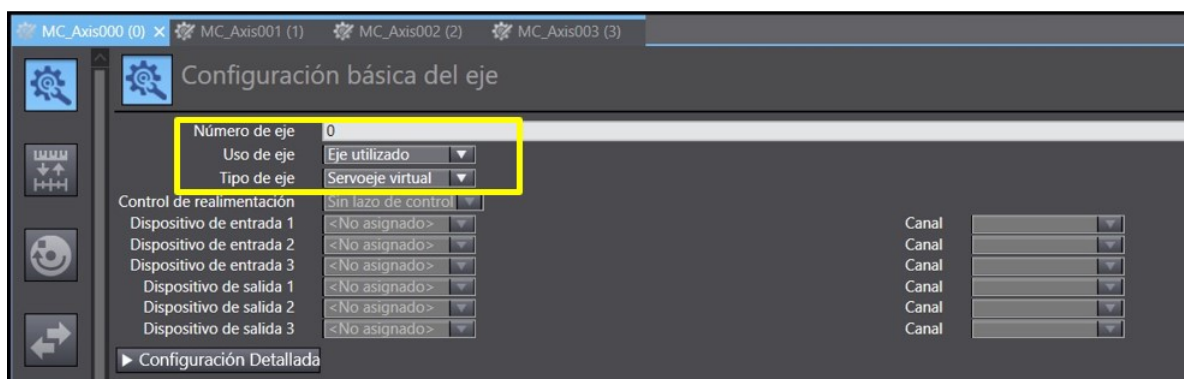


Figura 3.29: Configuración básica del eje.

En la ventana de **Configuración de conversión de unidades** se debe elegir la unidad con la que trabajará el eje. En el caso de los ejes A_0 , A_1 y A_3 al tener un movimiento rotacional la unidad elegida son los *grados*, figura 3.30. Sin embargo, el eje A_2 solo se mueve de arriba a abajo en el eje Z por lo que utiliza unidades de *mm*.

En la figura 3.30 también se ven los valores de las variables **contador de impulsos de comando por**

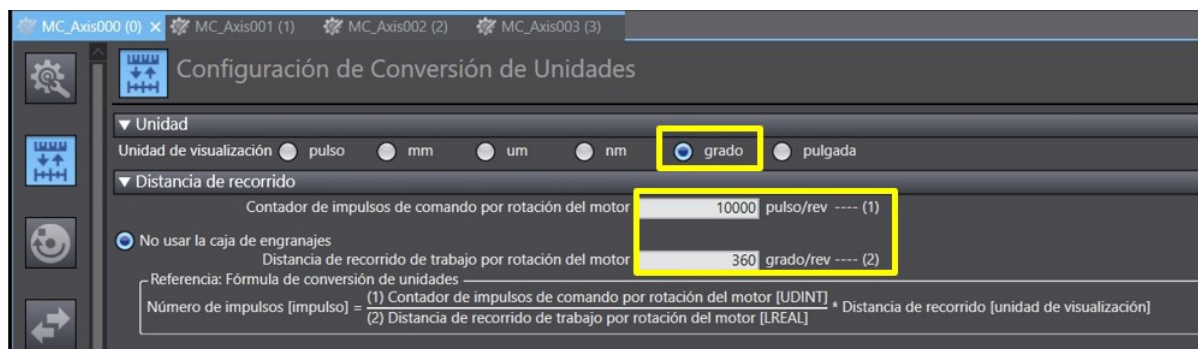


Figura 3.30: Configuración de Conversión de unidades.

rotación del motor y distancia de recorrido de trabajo por rotación del motor, que se eligieron viendo las especificaciones del robot SCARA (cita).

Por último, los últimos parámetros que se necesitan ajustar están en la ventana de **configuración de operación**, donde se especifica el valor máximo de velocidad que puede alcanzar el eje, cuyo valor también se encuentra en las especificaciones del robot, figura 3.31.

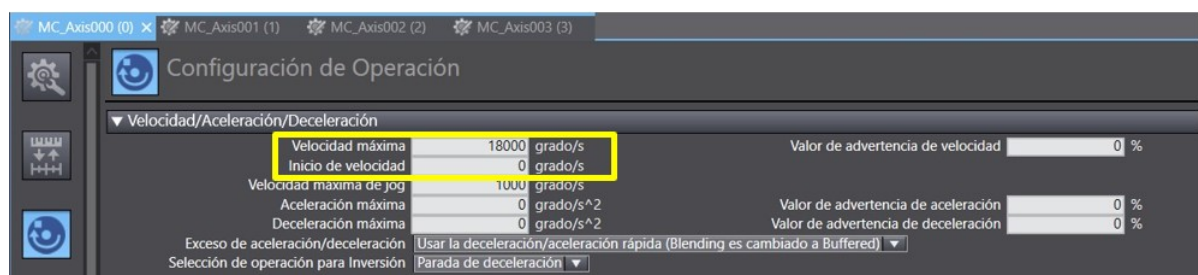


Figura 3.31: Configuración de operación.

Una vez que los ejes están configurados se debe configurar el robot entero asignando a cada eje del robot los ejes individuales que se han creado, para ello se accede a **configuración de grupo** → **Añadir** → **configuración de grupo de ejes**, figura 3.32.

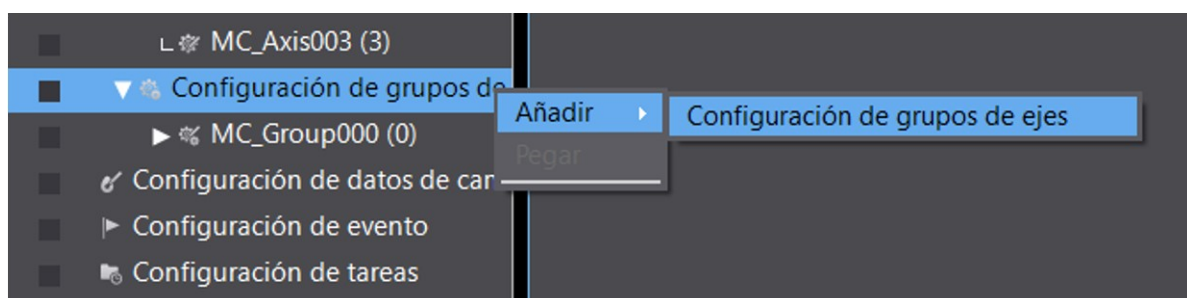


Figura 3.32: Creación grupo de ejes.

En la figura 3.33 se muestra la forma de crear el grupo a partir de los ejes.

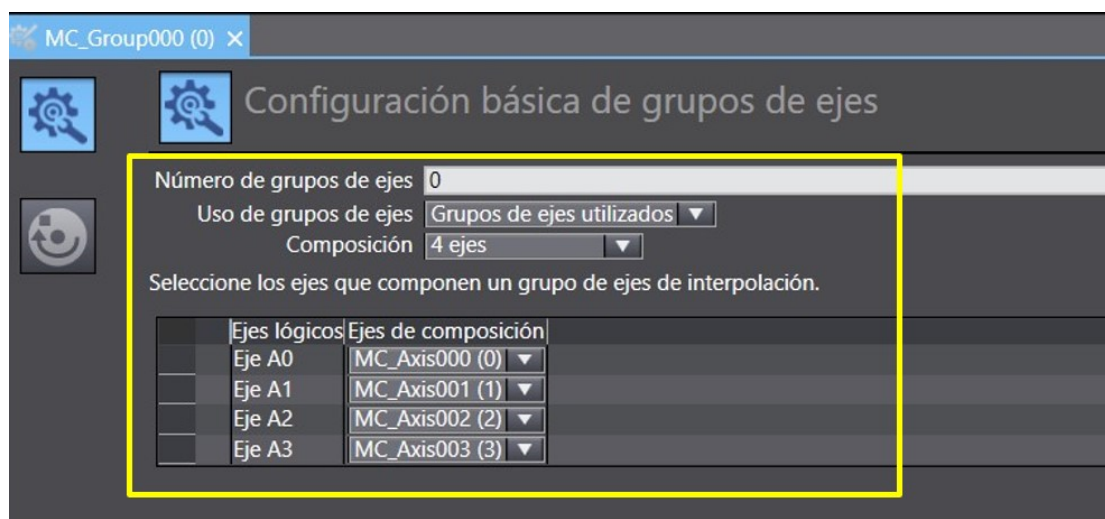
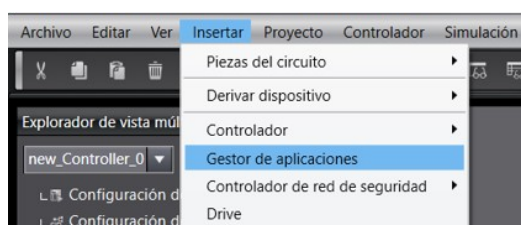


Figura 3.33: Configuración básica de grupo de ejes.

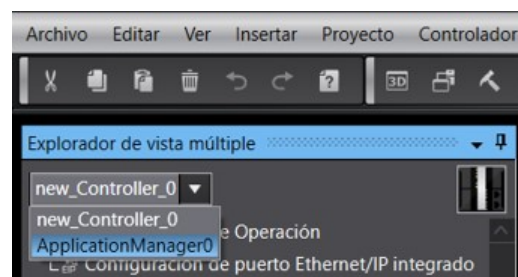
3.2.2 Creación de Datos de Forma 3D

Una vez que los ejes están configurados y se ha creado el grupo de ejes (robot completo) se pasa al segundo paso que es la creación del robot SCARA en 3D.

En primer lugar se debe agregar al proyecto el **gestor de aplicaciones**, que es un dispositivo lógico que administra los datos y la configuración necesaria para utilizar la función de simulación 3D, figura 3.34.



(a) Creación del gestor de aplicaciones.



(b) Ventana de gestor de aplicaciones.

Figura 3.34: Creación de un Robot SCARA utilizando Sysmac Studio 3D.

Una vez que se accede al gestor de aplicaciones se pulsa con el botón derecho sobre **visualización 3D** → **Añadir** → **componente mecánico**, figura 3.35, y se añaden los componentes mecánicos que forman el robot SCARA. Cada componente se añade a través de un fichero de tipo CAD que tiene extensión *.step*. Los ficheros tipo CAD de partes de un robot se pueden encontrar en múltiples páginas de internet, en este caso se cogieron de *GRABCAD*

Son cuatro los componentes mecánicos que se deben añadir al proyecto: el brazo, el antebrazo, el Eje Z y el Eje R, figura 3.24. Estos componentes mecánicos se deben asociar a los cuatro ejes creados en la sección 3.2.1 como se muestra en la figura 3.36. Como la base no realiza un movimiento lineal, ni rotacional no debe estar ligada a ningún eje por lo que no se agrega como un componente mecánico si no que se añade como un caja que tiene incorporado el propio simulador.

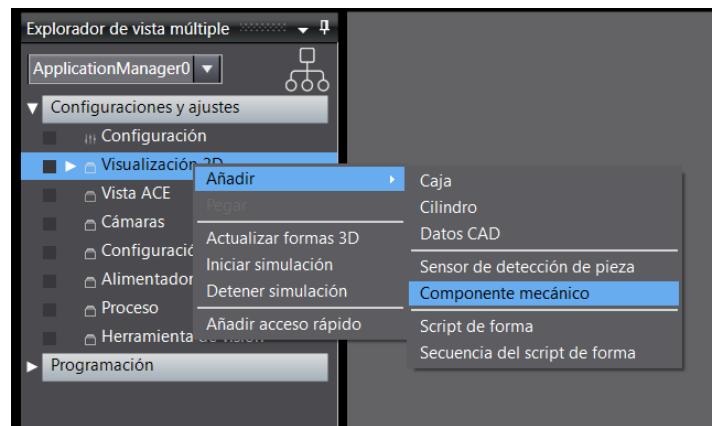


Figura 3.35: Añadir un componente mecánico.

Componente mecánico	Ejes
Brazo	Eje 0: MC_Axis000 (0)
Antebrazo	Eje 1: MC_Axis000 (1)
Eje Z	Eje 2: MC_Axis000 (2)
Eje R	Eje 3: MC_Axis000 (3)

Figura 3.36: Relación entre componente mecánico y eje del robot.

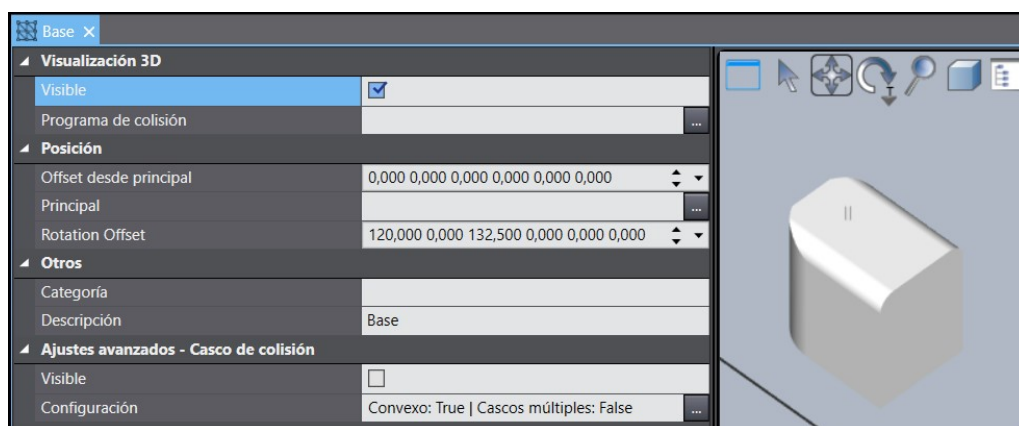


Figura 3.37: Configuración de la BASE del robot SCARA.

BASE

La base se carga como una caja que ofrece el simulador y se posiciona en un punto arbitrario del espacio, figura 3.37.

BRAZO

El brazo, como se ha visto, se añade como un componente mecánico donde hay que configurar algunos parámetros. En primer lugar en la ventana **configuración de parámetros** se escoge el eje con el que será asociado, figura 3.38.

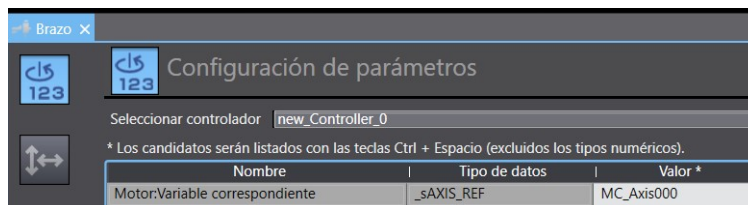


Figura 3.38: Configuración de parámetros del BRAZO del robot SCARA.

Como el *eje 0*, al que va asociado el brazo, hace un movimiento de rotación se debe configurar la ventana de **configuración de eje de rotación**, figura 3.39.

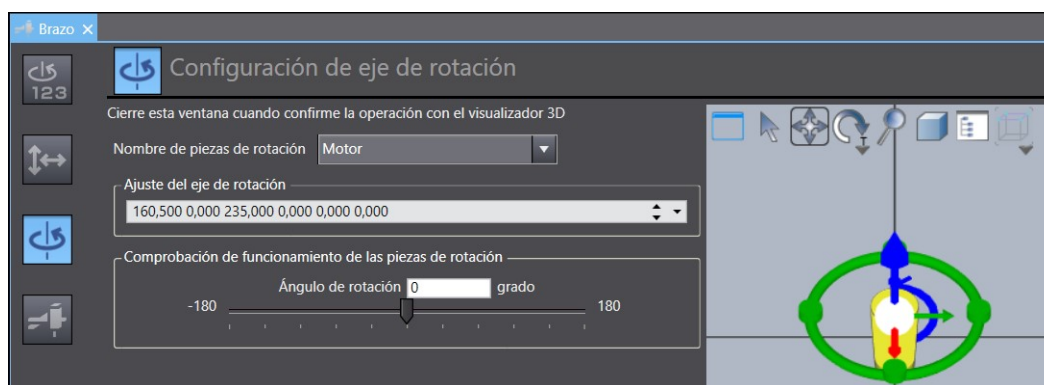


Figura 3.39: Configuración de eje de rotación del BRAZO del robot SCARA.

Por último, en la ventana de **configuración común de los componentes mecánicos** se debe especificar el parámetro **principal**, que indica a que otro componente va anclado. En este caso el brazo debe ir anclado a la base, figura 3.40.

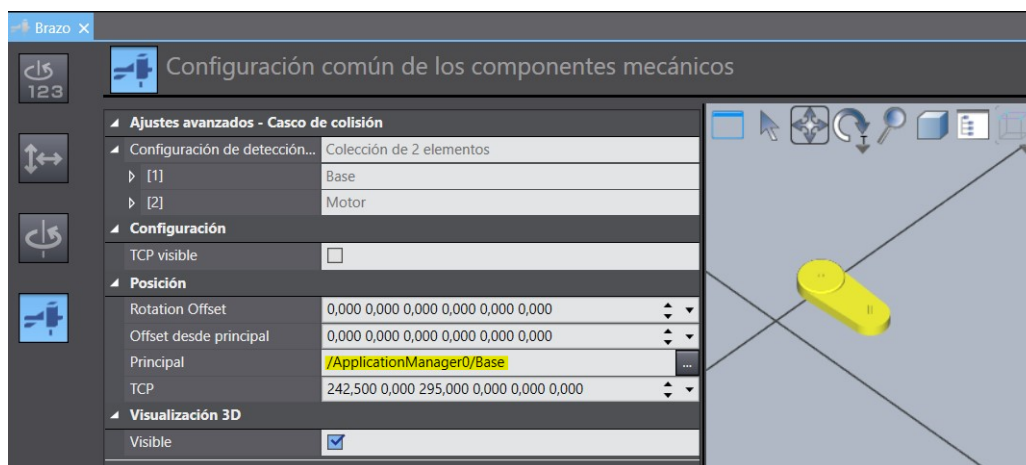


Figura 3.40: Configuración común del BRAZO del robot SCARA.

ANTEBRAZO

Como en el caso del brazo, en la ventana **configuración de parámetros** se escoge el eje con el que será asociado, figura 3.41.

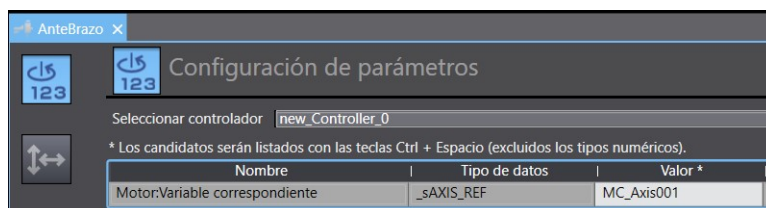


Figura 3.41: Configuración de parámetros del ANTEBRAZO del robot SCARA.

El antebrazo también realiza un movimiento rotacional por lo que se configura igual que el brazo con la única diferencia de que, en este caso, el antebrazo debe estar anclado al brazo, figura 3.42.

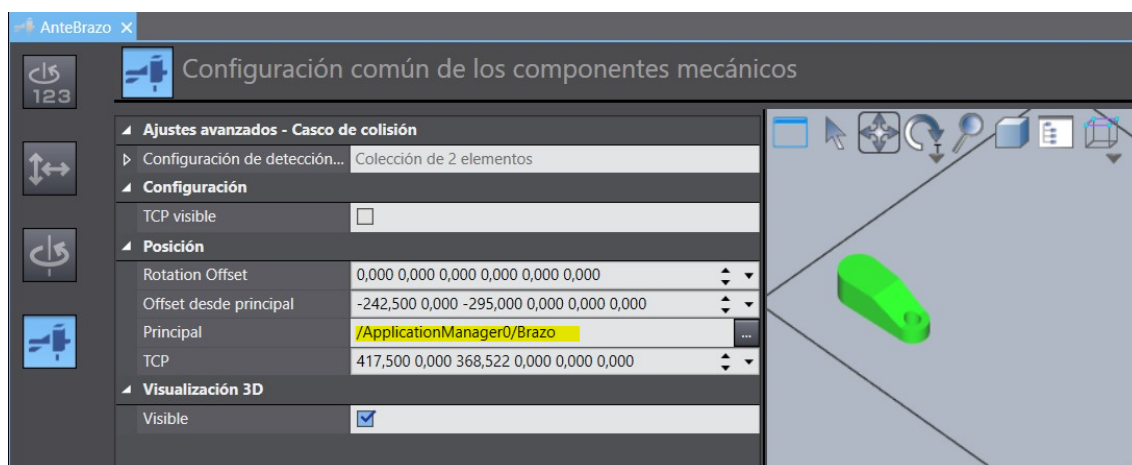


Figura 3.42: Configuración común del ANTEBRAZO del robot SCARA.

EJE Z

Como en los dos casos anteriores, en la ventana **configuración de parámetros** se escoge el eje con el que será asociado, figura 3.43.

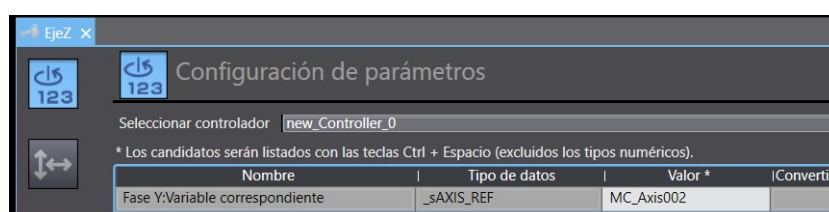


Figura 3.43: Configuración de parámetros del EJE Z del robot SCARA.

En este caso el *EJE Z* no realiza un movimiento rotacional sino lineal, ya que se mueve de arriba a abajo. En este caso se debe configurar la ventana de **configuración de dirección lineal**, figura 3.44.

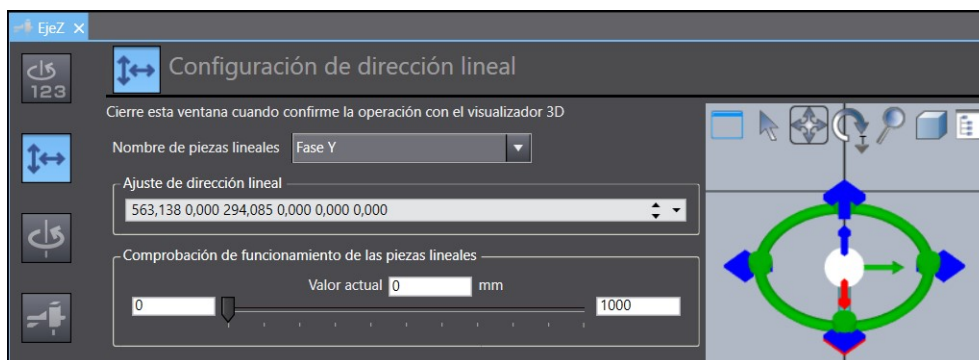


Figura 3.44: Configuración de dirección lineal del EJE Z del robot SCARA.

Por último, se configura el parámetro **principal** para que el *EJE Z* vaya anclado al antebrazo, figura 3.45.

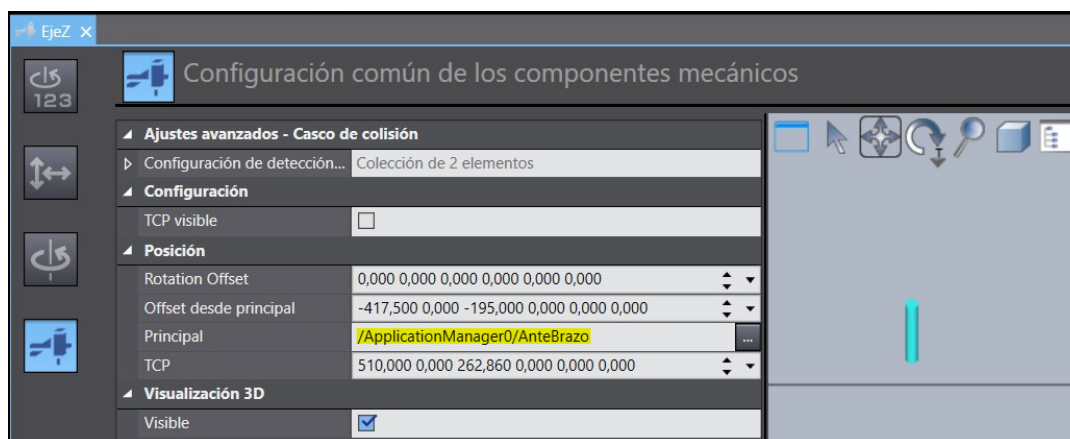


Figura 3.45: Configuración común del EJE Z del robot SCARA.

EJE R

Por último, para el *EJE R* se sigue el mismo procedimiento. En la ventana **configuración de parámetros** se escoge el eje con el que será asociado, figura 3.46.

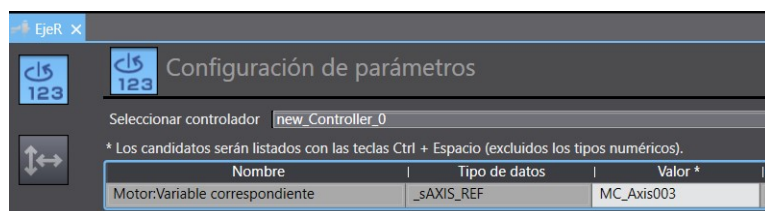


Figura 3.46: Configuración de parámetros del EJE R del robot SCARA.

El *EJE R* también realiza un movimiento rotacional por lo que se configura como en el caso del *brazo*. Para terminar, se configura el parámetro **principal** para que el *EJE R* vaya anclado al *EJE Z*, figura 3.47.

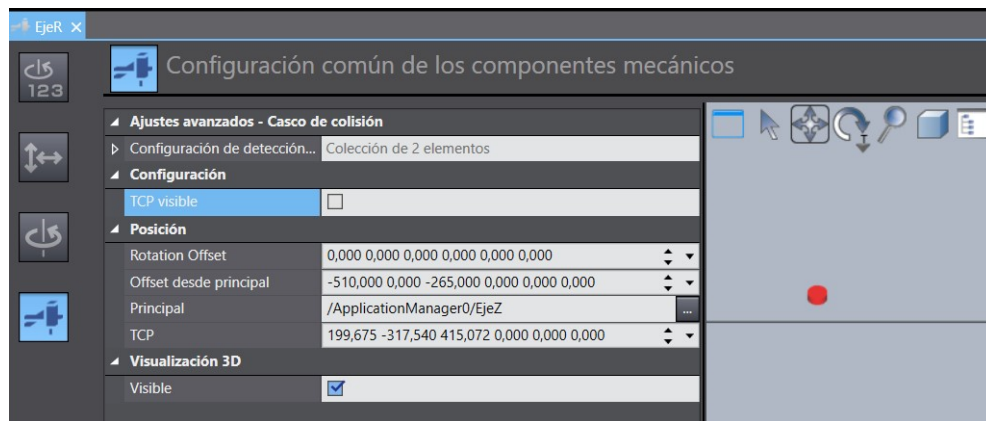


Figura 3.47: Configuración común del EJE Z del robot SCARA.

3.2.3 Creación del programa para el movimiento del Robot

Una vez creado el Robot SCARA en 3D el último paso es crear un programa con bloques de funciones del módulo de Robótica para generar el movimiento del robot.

Para ello, se explican a continuación los bloques de funciones que se van a utilizar y en el apartado 4.2 se verá detalladamente la forma de crear el programa para el movimiento del robot.

1. **MC_Power**: Este bloque de función se encarga de proporcionarle energía a los ejes, figura 3.48.

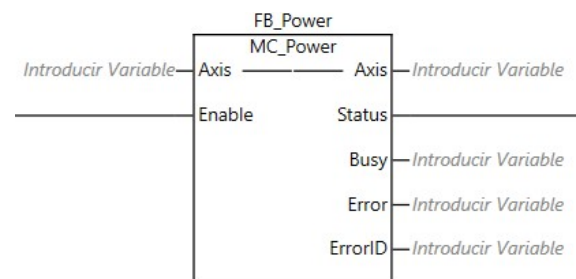


Figura 3.48: Bloque de función: MC_Power en Ladder.

2. **MC_GroupEnable**: Este bloque de función se encarga de habilitar el grupo de ejes, figura 3.49.

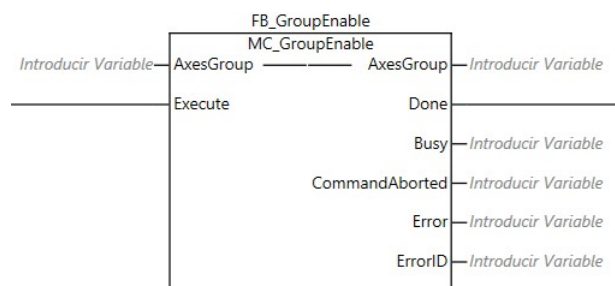


Figura 3.49: Bloque de función: MC_GroupEnable en Ladder.

3. **MC_SetKinTransform**: Este bloque de función establece la cinemática del robot, el área de trabajo para un grupo de ejes, y la velocidad de interpolación, figura 3.50. Es necesario para su

ejecución que el grupo de ejes esté deshabilitado. Debe establecerse cada vez que se da alimentación al controlador NJ, aunque, se mantiene al pasar al modo *PROGRAM*.

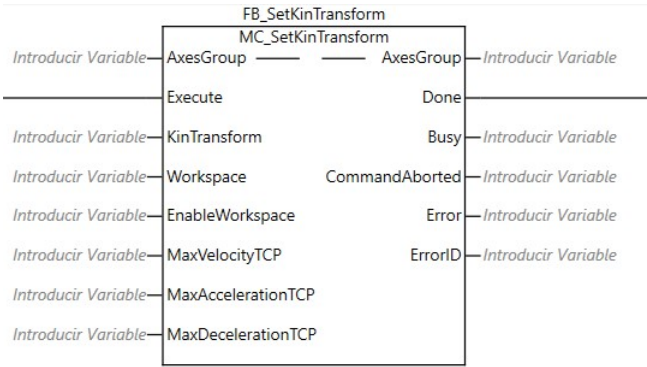


Figura 3.50: Bloque de función: MC_SetKinTransform en Ladder.

En el anexo A se muestra el tipo de datos de las variables de entrada del bloque de función *MC_SetKinTransform* y el rango de valores que se puede introducir.

4. **MC__MoveTimeAbsolute**: Este bloque de función crea variables de control para mover el robot a una posición específica en un tiempo determinado. Es importante tener en cuenta que no comprueba que el robot haya llegado a la posición objetivo, sino que la instrucción finaliza en el momento en que se ha cumplido el tiempo especificado, figura 3.51.

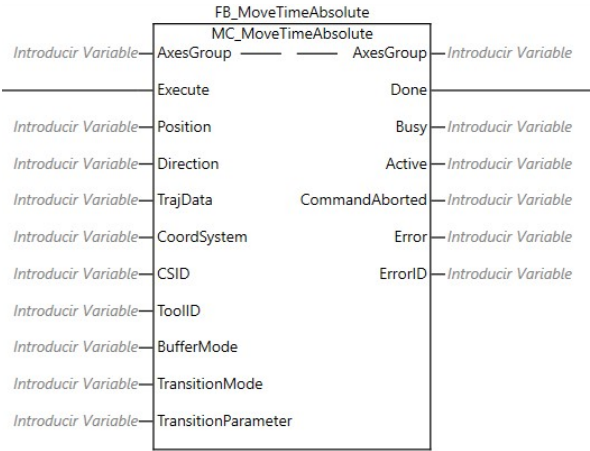


Figura 3.51: Bloque de función: MC_MoveTimeAbsolute en Ladder.

En el anexo B se muestra el tipo de datos de las variables de entrada del bloque de función *MC_MoveTimeAbsolute* y el rango de valores que se puede introducir.

Los bloques de funciones del módulo de *Robótica* son cajas negras para el usuario, es decir, Sysmac no permite acceder a la programación interna de cada bloque, esto puede suponer una desventaja si se quisiera comprender en su totalidad el movimiento del robot, pero, de cara a las industrias, es un sistema con grandes ventajas, ya que la programación es sencilla y fácil de aprender.

Capítulo 4

Resultados obtenidos

En este capítulo se muestran los resultados obtenidos al realizar las dos aplicaciones descritas en el capítulo 3.

4.1 Resultados de Trazabilidad

En el apartado 3.1 se explicaron detalladamente los pasos que se deben seguir para detectar, con el sensor de visión, piezas en la cinta transportadora y mandar los datos de salida al controlador. En el subapartado 3.1.3 se explicaron los cuatro métodos de detección diferentes para la detección de productos por la línea de fabricación. En este apartado se verá la aplicación práctica de dos de ellos:

1. **Búsqueda de forma.**
2. **Etiquetado.**

Se han elegido estos métodos por su gran aplicación en la industria y por la sencillez de las piezas a utilizar. Si nos enfocamos en las industrias de cosmética, el método de detección por excelencia es el de **Posición por bordes** ya que puede detectar si las colonias se encuentra bien posicionadas dentro de los godets.

El flujo de pasos a seguir para el análisis de los dos métodos de detección elegidos se pueden ver en la figura 4.1. A continuación, se analizan ambos métodos.

4.1.1 Búsqueda de forma

Después de realizar los dos primeros pasos descritos en los apartados 3.1.1, conexión con la máquina, y 3.1.2, calibración del sensor, se llega al paso de elegir el método de detección que se va a utilizar. En este primer apartado se realizará una prueba de detección de piezas con el método de **búsqueda de forma** siguiendo los pasos de la figura 4.1.

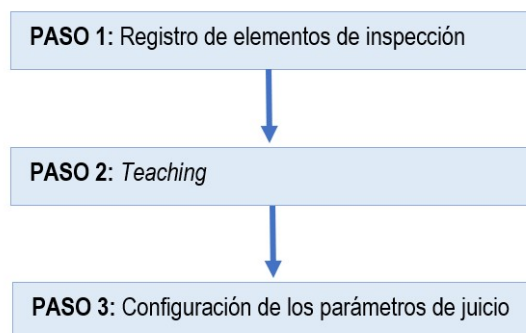


Figura 4.1: Pasos para la configuración del método de detección

PASO 1: REGISTRO DE ELEMENTOS DE INSPECCIÓN

Se escoge de la ventana de herramientas el método de detección que se va a utilizar y se arrastra a la ventana de **inspección**, como se observa en la figura 4.2. Para poder ver en la escena del simulador lo que está viendo el sensor de visión se debe poner el programa en **modo RUN**, figura 4.3.

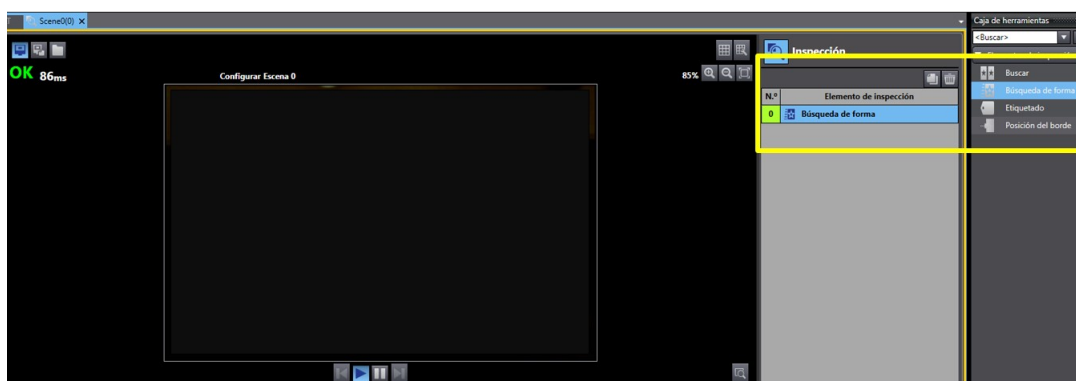


Figura 4.2: Registro del elemento de inspección.

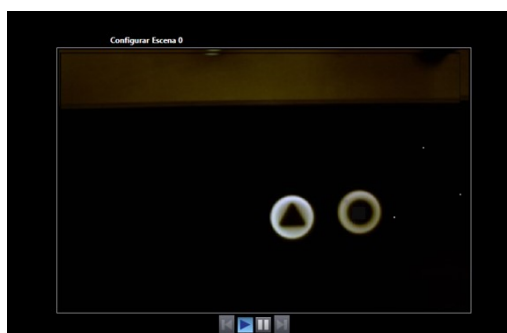


Figura 4.3: Reproducción de la escena del sensor de visión en el simulador. Modo RUN.

PASO 2: *TEACHING*

Teaching significa almacenar la región y la imagen como datos de referencia para la medición. Es decir, en este apartado se escoge el modelo que se quiere detectar y la región de medida donde el sensor de

visión puede detectar las piezas iguales al modelo.

Para escoger el modelo hay que pulsar, en la pantalla de inspección, el método de inspección escogido y configurar el modelo en la pestaña **región del modelo**, como se muestra en la figura 4.4.

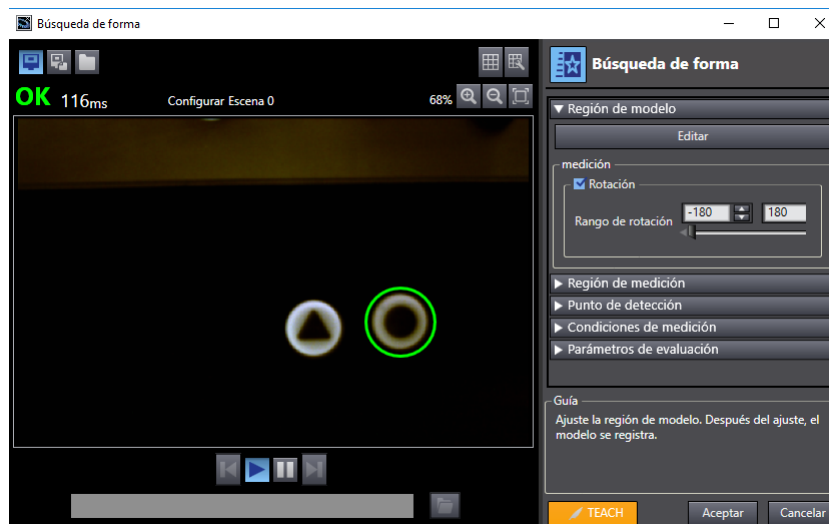


Figura 4.4: Registrar la región del modelo.

Como se ve en la imagen 4.4 se puede escoger el ángulo de rotación, lo que quiere decir que, se puede configurar si las piezas a detectar por el sensor pueden ir rotadas con respecto al modelo o no. En este caso se ha elegido una rotación de -180° a 180° por lo que las piezas pueden ir 180° rotadas con respecto al modelo y, aun así, pueden ser detectadas.

Un vez configurada la región del modelo se debe establecer la región de medida, el área donde el sensor debe buscar las piezas iguales al modelo, esto se hace en el apartado de **región de medición**, figura 4.5.

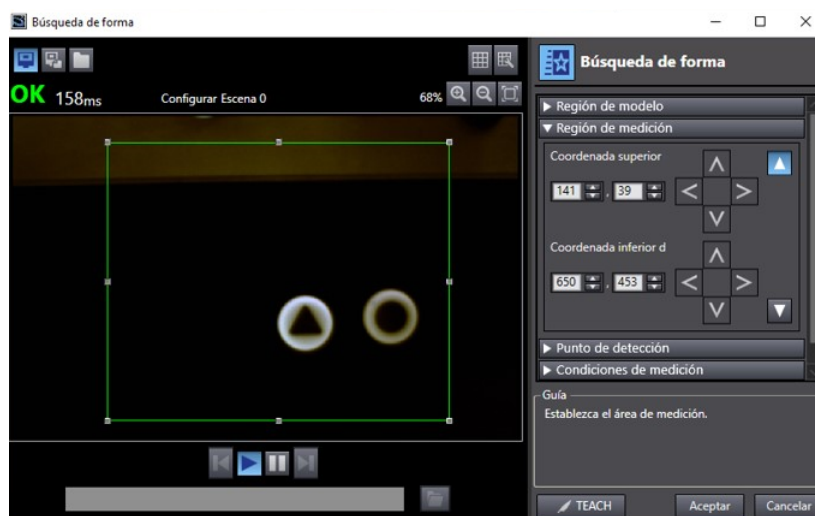


Figura 4.5: Determinar la región de medición.

PASO 3: CONFIGURACIÓN DE LOS PARÁMETROS DE JUICIO

En este paso se establecen los rangos de algunos parámetros que se consideran correctos a la hora de detectar las piezas en la cinta.

En primer lugar se ajusta el **Punto de detección** donde se puede especificar qué parte del modelo detectar como coordenada durante la inspección. Normalmente, el centro del modelo registrado se utiliza como punto de detección. Se puede ver en la figura 4.6.

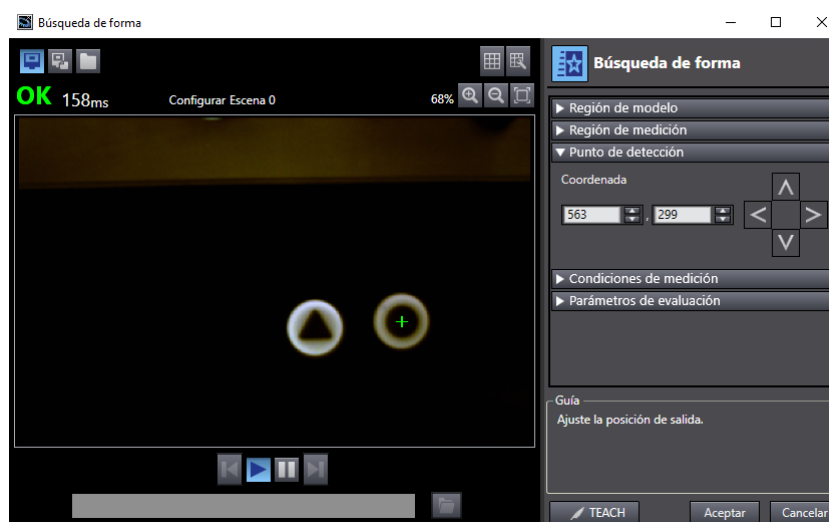


Figura 4.6: Determinar el punto de detección.

En segundo lugar se configuran los parámetros de la ventana de **condiciones de medición**. En ella se puede escoger el valor de **correlación** que puede tener el modelo con la pieza a detectar. Si el nivel de correlación es demasiado bajo puede que las piezas detectadas sean muy diferentes al modelo. En la figura 4.7 se escoge un nivel de correlación de 76. Si nos fijamos en la escena, la pieza con forma de triángulo es detectada, lo que supone un error en la detección de las piezas. Para este caso, el nivel de correlación se puso en 98 para que únicamente la piezas con forma de círculo fueran detectadas, 4.8.

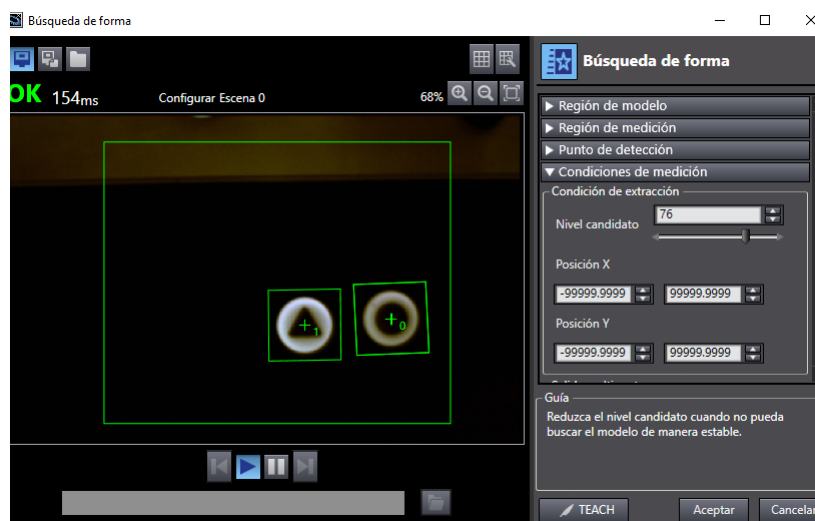


Figura 4.7: Nivel de correlación del modelo con las piezas de 76.

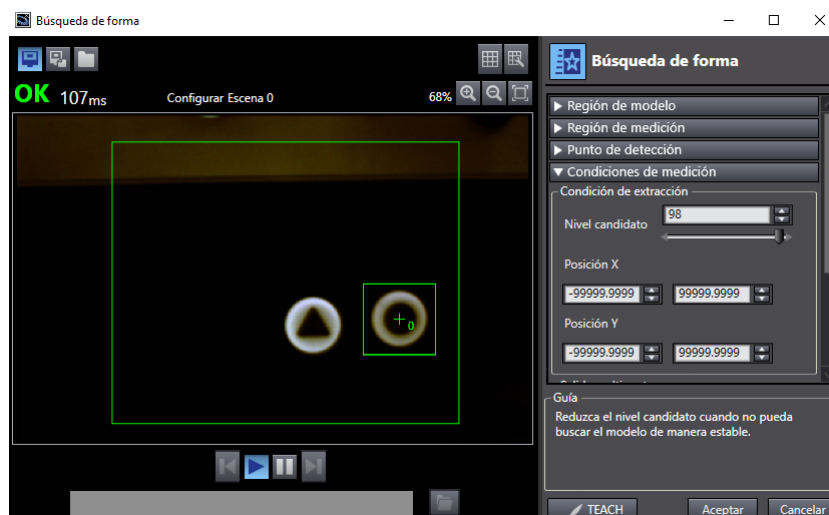


Figura 4.8: Nivel de correlación del modelo con las piezas de 98.

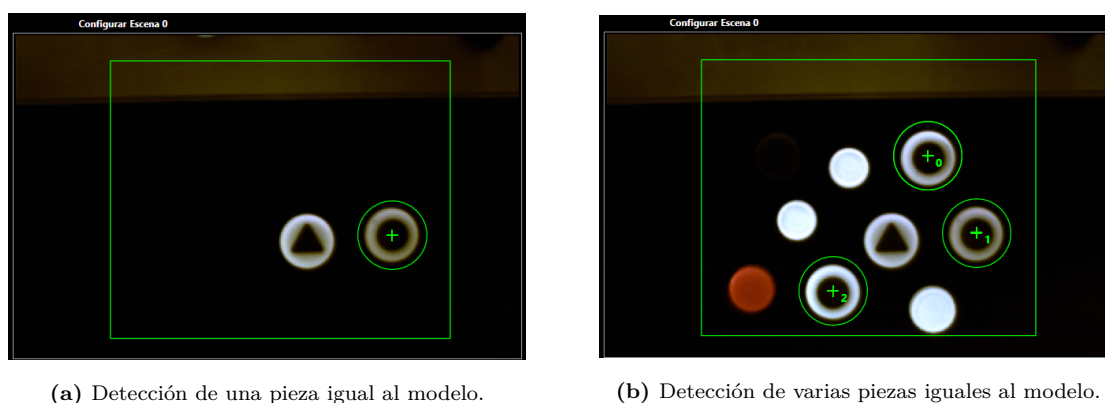


Figura 4.9: Detección de piezas mediante el método: BÚSQUEDA DE FORMA.

En las figuras 4.7 y 4.8 también se pueden configurar los parámetros **Posición X** y **Posición Y** para que detecte las piezas cuyas posiciones X e Y estén dentro de esos rangos. Para este caso se deja el valor por defecto.

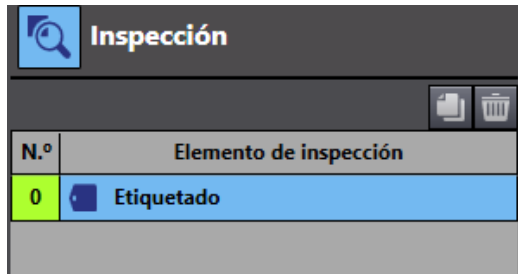
Cuando todos los parámetros están configurados se le da al botón de aplicar y la pieza que se ha utilizado para crear el modelo es detectada, figura 4.9a. Si se añaden más piezas iguales al modelo deben ser detectadas como ocurre en la figura 4.9b.

4.1.2 Etiquetado

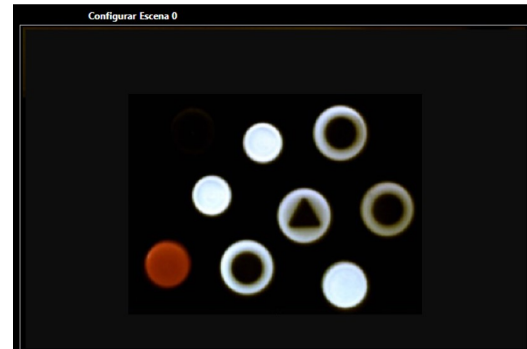
De la misma forma que para el caso anterior, apartado 4.1.1, lo primero que se debe hacer es la conexión con la máquina, apartado 3.1.1 y la calibración del sensor de visión, apartado 3.1.2. En este caso se explica el método de detección de **Etiquetado** siguiendo los pasos de la figura 4.1.

PASO 1: REGISTRO DE ELEMENTOS DE INSPECCIÓN

En primer lugar se registra el elemento de inspección que se quiere utilizar para configurar el modelo, figura 4.10a. A continuación, se pone el sistema en **modo RUN** para detectar lo que está viendo el sensor de visión, figura 4.10b



(a) Registro del elemento de inspección.



(b) Reproducción de la escena del sensor de visión en el simulador. Modo RUN

Figura 4.10: Registro del elemento de inspección: Etiquetado

PASO 2: *TEACHING*

En el método de detección por **Etiquetado** como se explicó en el apartado 3.1.3, se detectan cuantos grupos de color existen dentro de una región de medida. En este paso se debe seleccionar qué color es que el que quiere detectar y la región de medida donde las piezas pueden ser detectadas por el sensor.

En las figuras 4.11 y 4.12 se seleccionan los píxeles de una ficha roja para ser el color que, más tarde, el sensor de visión debe detectar.

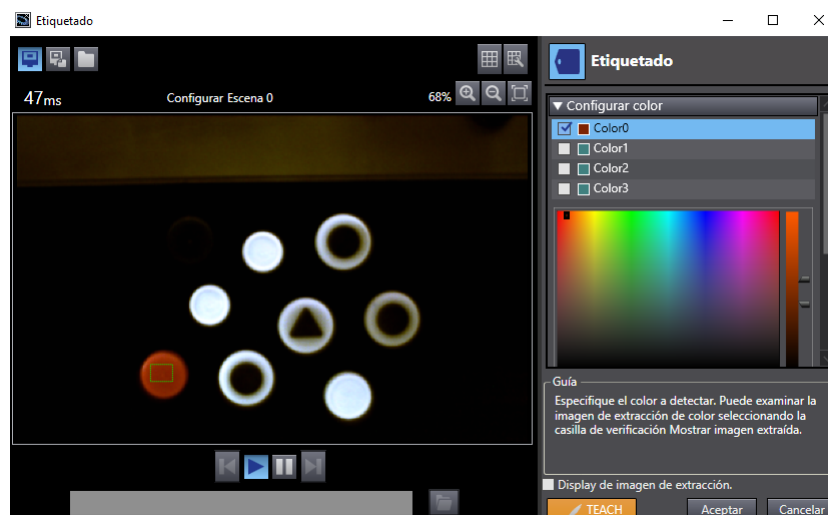


Figura 4.11: Selección de píxeles de una pieza roja.

En la figura 4.13 se selecciona la región de medición donde el sensor debe detectar las piezas iguales al modelo.

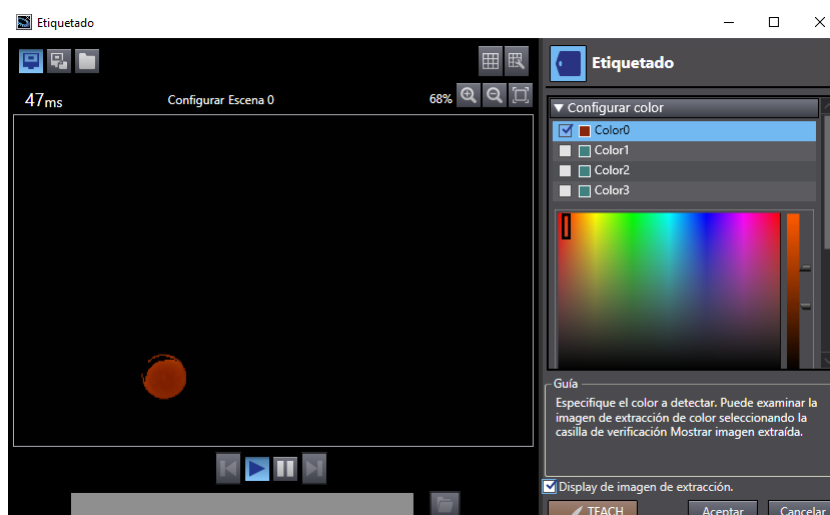


Figura 4.12: Selección de píxeles de una pieza roja.

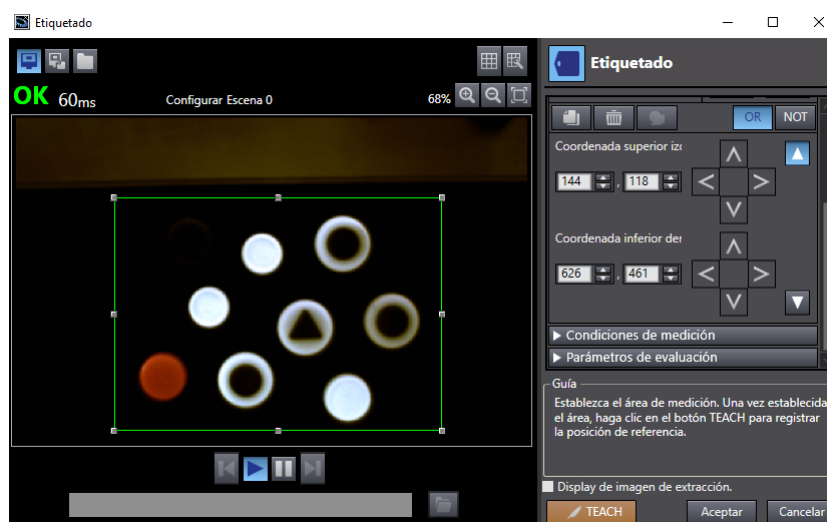


Figura 4.13: Determinar la región de medición.

Finalmente se pulsa el botón *teach* para guardar los cambios y pasar al siguiente paso.

PASO 3: CONFIGURACIÓN DE LOS PARÁMETROS DE JUICIO

Por último, se configuran las **condiciones de medición**. Para este método los parámetros más importantes son el **rellenado de orificios** y **eliminar la duplicidad**. El relleno de orificios establece cómo procesar las áreas rodeadas por el color especificado. Si esta casilla de verificación está seleccionada, esas áreas se procesan como el color especificado. Para este caso, como la pieza es toda de color rojo elegimos la opción de relleno de orificios, 4.14.

Si se pulsa el botón de aplicar y se colocan más piezas de color rojo sobre la región de medida, el sensor detecta las piezas y las enumera colocando una etiqueta sobre ellas. En la figura 4.15 el sensor coloca dos etiquetas en una de las piezas, para que eso no ocurra se debe seleccionar la opción de **eliminar la duplicidad**, figura 4.16a. El resultado de la detección de las piezas rojas se observa en la figura 4.16b.



Figura 4.14: Condiciones de medida: rellenado de orificios.

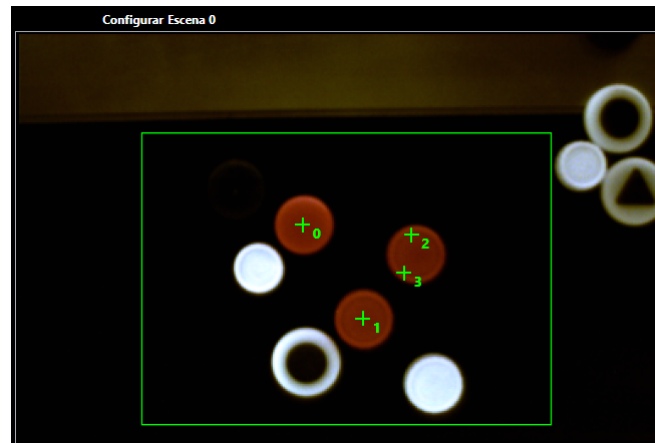
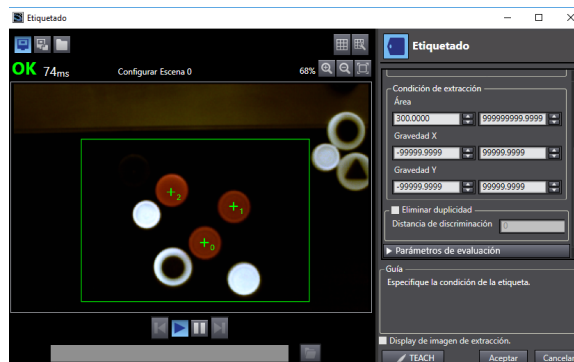
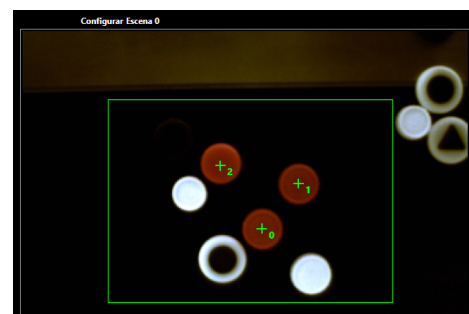


Figura 4.15: Detección de una pieza con do etiquetas.



(a) Condición de medida: Eliminar duplicidad.



(b) Resultado de la detección de piezas rojas.

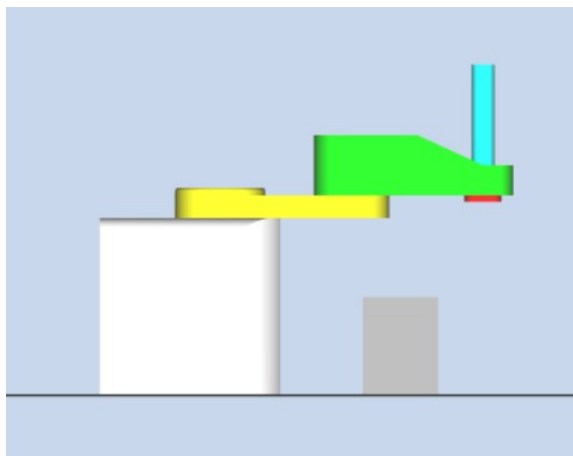
Figura 4.16: Detección de piezas mediante el método: ETIQUETADO

4.2 Resultados de Pick&Place

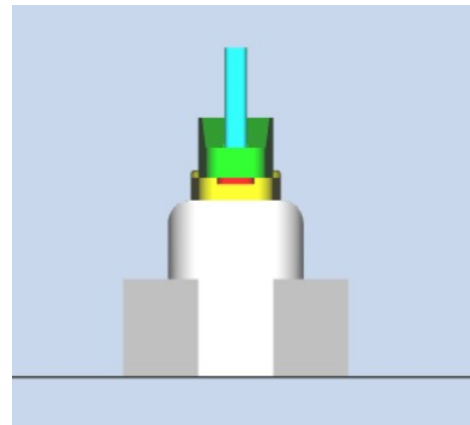
En este último apartado se muestra el resultado del montaje completo del robot SCARA en 3D y el programa, formado por los bloques de funciones explicados en la sección 3.2.3, para el movimiento de Pick&Place del robot.

4.2.1 Montaje completo Robot SCARA 3D

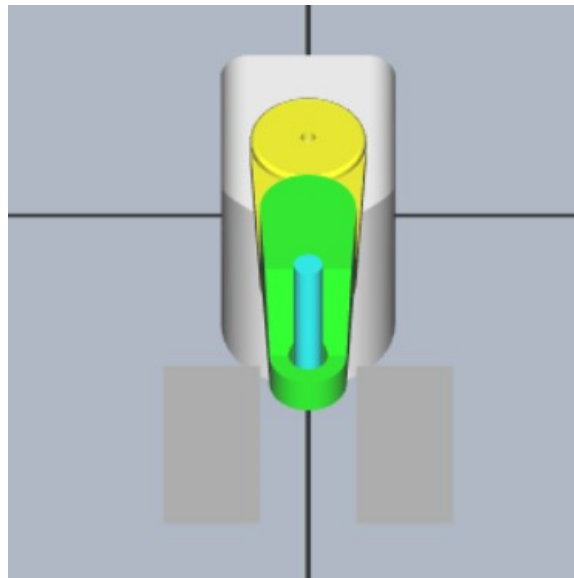
El montaje del Robot completo se puede ver en la figura 4.17 desde distintas perspectivas.



(a) Robot SCARA 3D de lado.



(b) Robot SCARA 3D de frente.



(c) Robot SCARA en 3D desde arriba.

Figura 4.17: Montaje del robot SCARA 3D con Sysmac Studio 3D.

4.2.2 Programa para el movimiento del ROBOT

Dentro de un programa, en Sysmac Studio, se pueden crear diferentes secciones que resultan muy útiles para organizar el movimiento del robot en los diferentes pasos que se deben seguir. En la figura 4.18 se muestran los pasos seguidos en la simulación del robot para generar el movimiento.

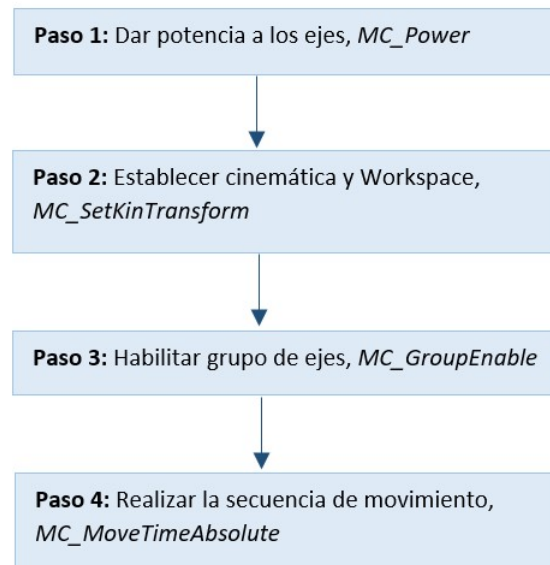


Figura 4.18: Pasos en la simulación para el movimiento del robot.

Para crear una sección en el programa principal se pulsa el botón derecho del ratón sobre **programa** → **Añadir** → **sección**.

Para el programa del movimiento del robot SCARA se han creado tres secciones como se observa en la figura 4.19.

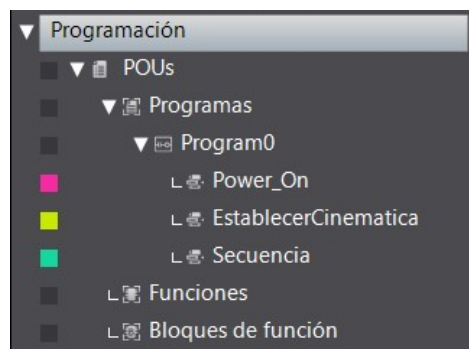


Figura 4.19: Secciones creadas en el programa principal para el movimiento del robot.

- **Sección Power_on:** Se utiliza el bloque de función *MC_Power* para dar energía a los cuatro ejes del robot. Corresponde con el *paso 1* de la figura 4.18.
- **Sección Establecer_Cinemática:** Se definen los parámetros necesarios para establecer la cinemática del robot y habilitar el grupo de ejes. Se corresponde con los *paso 2 y 3* de la figura 4.18.

- **Sección Secuencia:** Se utiliza una secuencia de bloques de función del tipo MC_MoveTimeAbsolute para generar el movimiento de Pick&Place del robot. Se corresponde con el último paso de la figura 4.18.

1. SECCIÓN Power_ON

El primer paso fundamental para que el robot puede hacer cualquier movimiento es dar energía a sus ejes. En este caso, como se vio en el apartado 3.2 un robot SCARA tiene cuatro ejes denominados *MC_Axis000* (0), *MC_Axis001* (1), *MC_Axis002* (2) y *MC_Axis003* (3). En la figura 4.20 se muestra la forma de conectar los ejes con el bloque de función *MC_Power*.

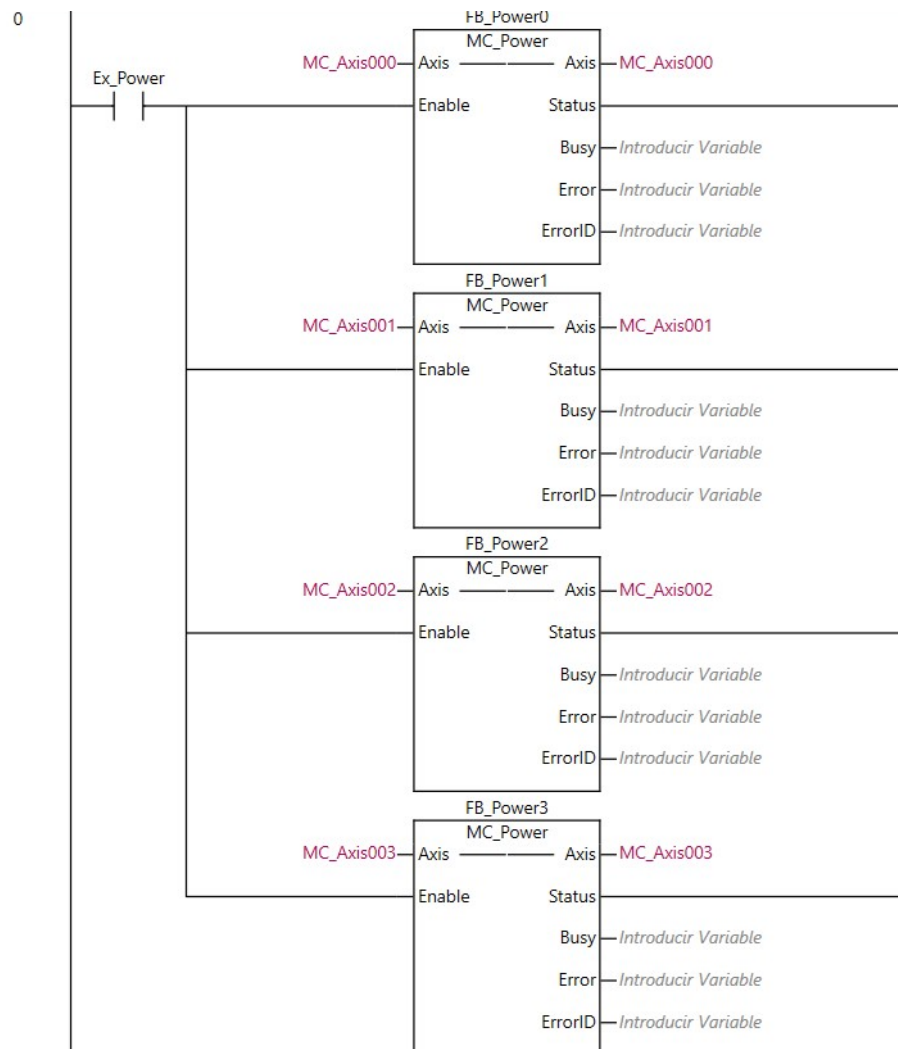


Figura 4.20: Sección Power_ON → Paso 1.

Para esta sección se ha creado una variable interna *Ex_Power* de tipo *BOOL* que cuando vale *TRUE* da energía a los ejes y cuando vale *FALSE* les quita dicha energía.

2. SECCIÓN Establecer__Cinemática

Es importante establecer la cinemática y el espacio de trabajo de un robot para que el simulador sepa qué movimiento debe generar cada eje. Una vez establecida la cinemática es necesario habilitar el grupo de ejes para que todo el robot pueda moverse en sincronía. En la figura 4.24 se muestra la sección *Establecer__cinemática* que consta de varias partes:

1. En la primera parte se utiliza texto estructurado para inicializar los parámetros necesarios para establecer la cinemática, el área de trabajo del robot, y la configuración de velocidad máxima de interpolación.
2. En la segunda parte se utiliza el bloque de funciones *MC_SetKinTransform* donde se utilizan los parámetros definidos en la parte anterior para establecer la cinemática, el área de trabajo y la velocidad de interpolación.
3. Por último, se habilita el grupo de ejes para poder trabajar con el robot entero utilizando el bloque de función *MC_GroupEnable*.

En la parte del texto estructurado lo primero que se debe hacer es indicar con qué tipo de robot se está trabajando. Para un robot de tipo SCARA RPR+R se define con el parámetro cinemático *KinType* el valor *_mcScaraRRPType2*. Después, se indican los parámetros de la cinemática del robot, que se vieron en el apartado 3.2, la velocidad, la aceleración y la deceleración de interpolación, como se muestra en la figura 4.21.

```

1  Cinematica.KinType:=_eMC_KIN_TYPE#_mcScaraRRPType2; //Tipo de Robot SCARA RPR+R
2  Cinematica.KinParam[0]:=175.0; // L1: Longitud del primer enlace
3  Cinematica.KinParam[1]:=175.00; //L2: Longitud del segundo enlace
4  Cinematica.KinParam[2]:=0.00; //d1: distancia desde el efector final a la base cuando A0=2.
5  Cinematica.KinParam[3]:=0.00; //Reservado
6  Cinematica.KinParam[4]:=0.00; //Reservado
7  Cinematica.KinParam[4]:=0.00; //Reservado
8  Cinematica.KinParam[5]:=0.00; //Reservado
9  Cinematica.KinParam[6]:=0.00; //Reservado
10 Cinematica.KinParam[7]:=0.00; //Reservado
11 MaxVTCP:=2000; //Velocidad máxima de interpolación
12 MaxATCP:=200000; // Aceleración máxima de interpolación
13 MaxDTCP:=MaxATCP; //Deceleración máxima de interpolación
14
15

```

Figura 4.21: Inicialización de los parámetros cinemáticos del robot SCARA en texto estructurado.

Además, se debe indicar el área de trabajo llamado en inglés *Workspace*. En este caso también es necesario indicar en primer lugar con qué tipo de robot se está trabajando y, después, definir los parámetros del workspace del robot, figura 4.22. En la figura 4.23 se muestra la programación del área de trabajo.

Una vez están todos los parámetros definidos se puede utilizar el bloque de función *MC_SetKinTransform* para establecer la cinemática del robot y *MC_GroupEnable* para habilitar el grupo de ejes, figura 4.24.

En ambos bloques de función, *MC_SetKinTransform* y *MC_GroupEnable*, se debe poner en la entrada *AxxGroup* el grupo de ejes creado *MC_Group000* y no los ejes independientes.

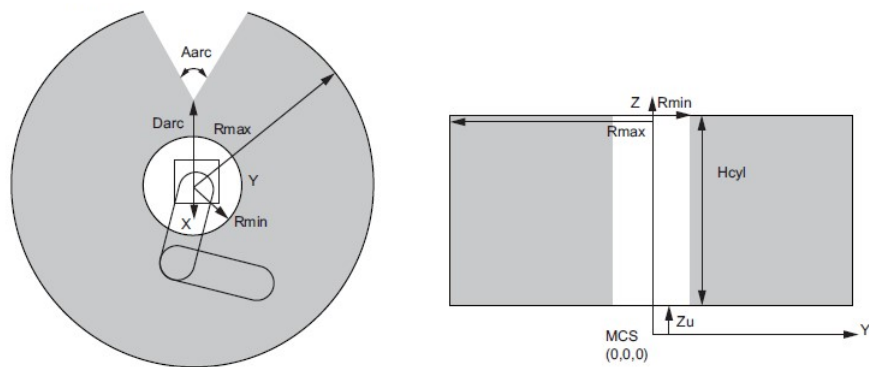


Figura 4.22: *Workspace* de un robot SCARA

```

16 Workspace.WorkspaceType:=_mcScaraWorkspace; //105 en manual de usuario (pag 128)
17 Workspace.WorkspaceParam[0]:=289.0; //Rmin: El radio interno del cilindro (mm)
18 Workspace.WorkspaceParam[1]:=1030.0; //Rmax: Radio externo del cilindro (mm)
19 Workspace.WorkspaceParam[2]:=1000.0; //Zu: Coordenada Z de la parte inferior del cilindro (mm)
20 Workspace.WorkspaceParam[3]:=1000.0; //Hcyl: Altura del cilindro (mm)
21 Workspace.WorkspaceParam[4]:=1000.0; //Darc: distancia del origen del arco (distancia desde el centro del
22 //cilindro hasta el origen del arco a lo largo de eje X negativo) (mm)
23 Workspace.WorkspaceParam[5]:=160.0; //Aarc: angulo del arco (grados)
24 Workspace.WorkspaceParam[6]:=0.00; //Reservado
25 Workspace.WorkspaceParam[7]:=0.00; //Reservado
26
27
28
29
30

```

Figura 4.23: Inicialización de los parámetros del *Workspace* del robot SCARA en texto estructurado.

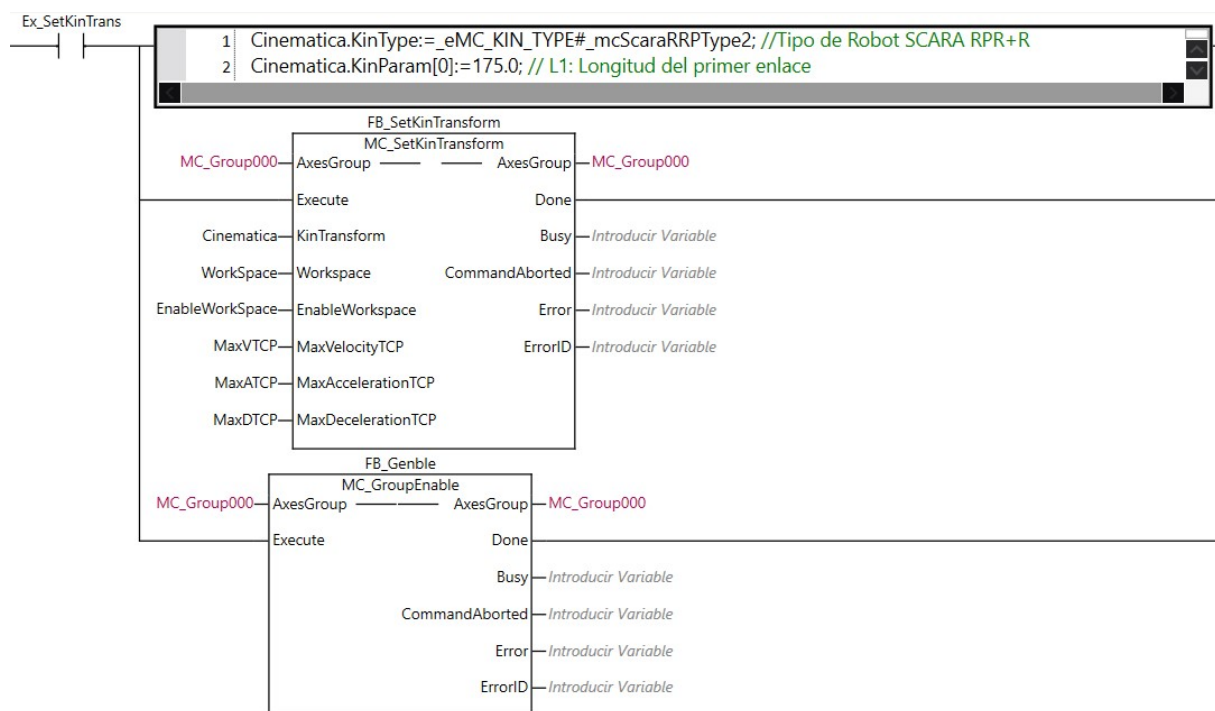


Figura 4.24: Sección Establecer_cinematica → Paso 2 y 3.

3. SECCIÓN Secuencia

Por último, en esta sección se genera el movimiento del robot. Para ello se utiliza una secuencia del bloque de función *MC_MoveTimeAbsolute*, figura 4.26. Este bloque de función, como se vio en el apartado 3.2.3, realiza movimientos punto a punto sin interpolación lineal (cada eje se mueve de forma independiente).

En primer lugar se inicializan las variables de entrada del bloque de función en texto estructurado, como se ve en la figura 4.25. Se ha elegido que se mueva en dirección positiva, con trayectoria de tipo *_mcPolynomial3*, tiempo objetivo de *1000 ms*, coordenadas de robot (UCS) y encadenando movimientos con *buffer*. Los valores de posición de destino establecidos con las variables de entrada Posición (*Posición de destino [X, Y, Z, Rx, Ry, Rz]*) varían según el tipo de robot seleccionado, en el caso del robot SCARA solo se utilizan las posiciones *X, Y, Z* y *Rz* por lo que se deben inicializar únicamente *Posición[0]*, *Posición[1]*, *Posición[2]* y *Posición[5]*.

```

6 Posicion2[0]:=250;
7 Posicion2[1]:=-100;
8 Posicion2[2]:=0;
9 Posicion2[5]:=0;
10
11 Posicion3[0]:=250;
12 Posicion3[1]:=-100;
13 Posicion3[2]:=85;
14 Posicion3[5]:=0;
15
16 Posicion4[0]:=250;
17 Posicion4[1]:=100;
18 Posicion4[2]:=0;
19 Posicion4[5]:=0;
20
21 Posicion5[0]:=250;
22 Posicion5[1]:=100;
23 Posicion5[2]:=85;
24 Posicion5[5]:=0;
25
26 Posicion6[0]:=350;
27 Posicion6[1]:=0;
28 Posicion6[2]:=0;
29 Posicion6[5]:=0;
30
31 Direccion[0]:=_eMC_DIRECTION#_mcNoDirection;
32 TrajData.MoveTrajType:=_mcPolynomial3;
33 TrajData.TrajTime:=1000;
34 CoordSystemMove:=_eMC_RBT_COORD_SYSTEM#_mcRBT_MCS;
35 BufferMode:=_eMC_BUFFER_MODE#_mcBuffered;
36 TransitionMode:=_eMC_TRANSITION_MODE#_mcTMNone;

```

Figura 4.25: Inicialización de los parámetros para el movimiento del robot SCARA.

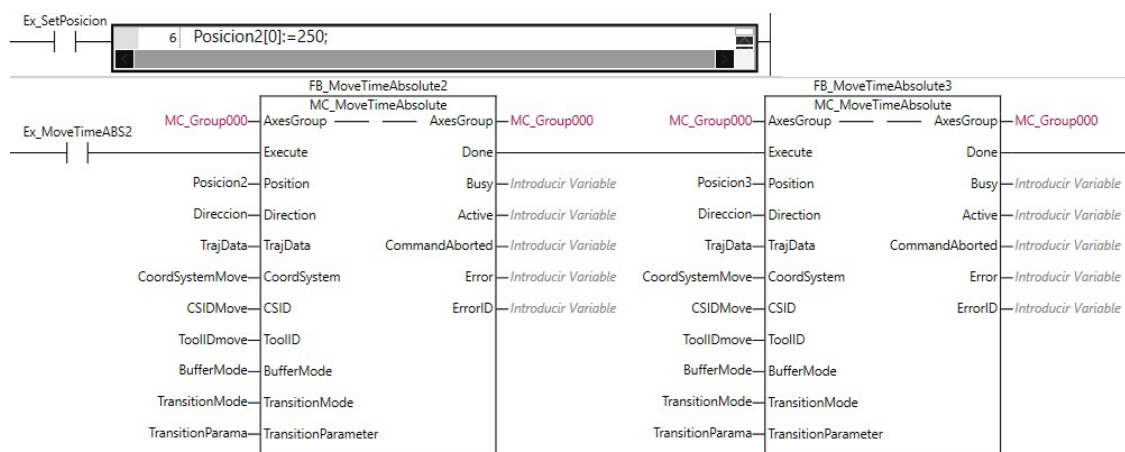


Figura 4.26: Sección Secuencia → Paso 4.

La secuencia de movimiento del Robot SCARA 3D se puede ver en la figura 4.27.

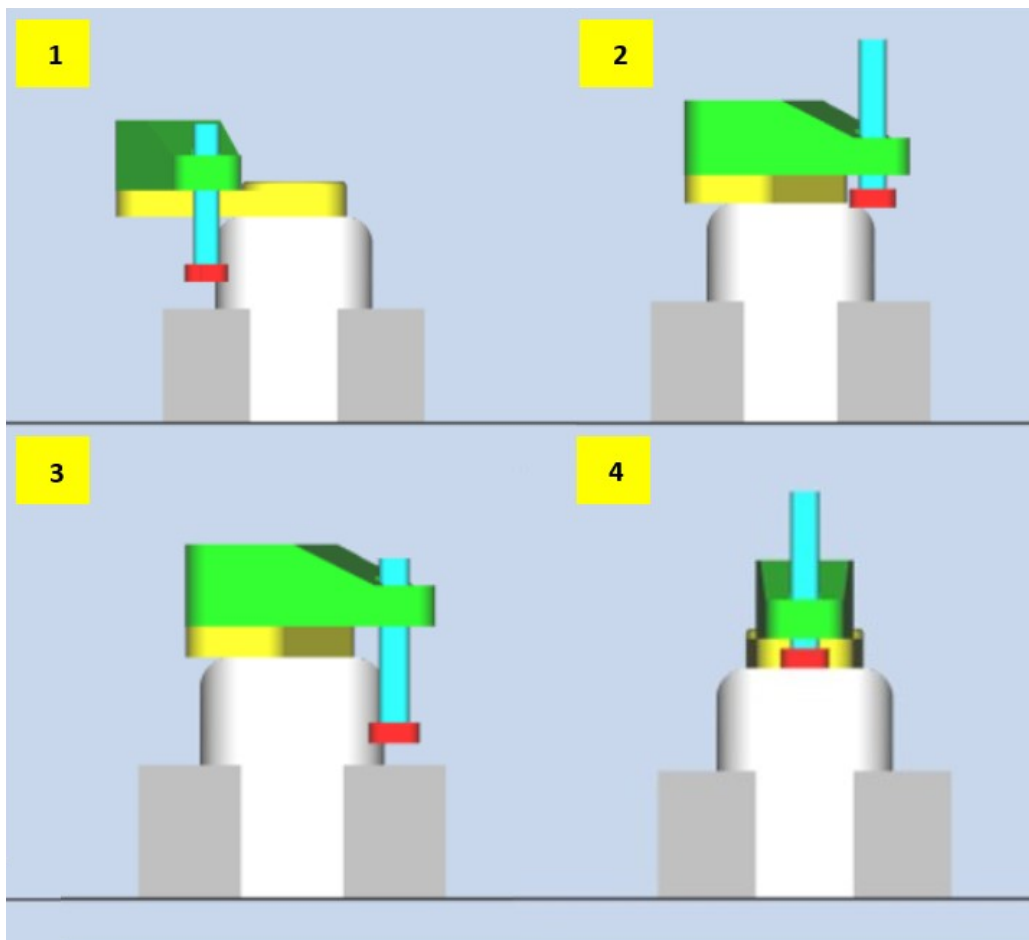


Figura 4.27: Movimiento de Pick&Place Robot SCARA RPR + R en 3D

Capítulo 5

Conclusiones y trabajos futuros

5.1 Conclusiones

Como bien dijo una vez Kazuma Tateisi, creador de la empresa Omron, *"Para la máquina, el trabajo de la máquina; para el hombre, el desafío de la creación"*. Automatizar procesos que son repetitivos y que pueden ser llevados a cabo por máquinas da una gran ventaja a la mente humana. El tiempo es nuestro peor enemigo, es limitado y no se puede comprar, es, por lo tanto, nuestro deber aprovecharlo al máximo. La automatización nos ayuda a invertir nuestro tiempo en pensar y en crear, y deja para las máquinas los trabajos redundantes con los que el humano no aprende nada nuevo. En las industrias apostar por sistemas automatizados reduce costes, aumenta la calidad de los productos, el bienestar físico y mental de los trabajadores y, en general, aumenta la productividad y la competitividad de las empresa.

Los objetivos que se comentaron en la introducción de este proyecto, apartado 1.2, se han cumplido satisfactoriamente. En primer lugar, se ha realizado un estudio extenso sobre los conceptos de *Automatización* y *Trazabilidad*, y la importancia de implementar sus soluciones en las industrias, más concretamente, en las industrias de cosmética. Añadido a lo anterior se han descrito algunos dispositivos que la empresa Omron ofrece para llevar a cabo sistemas automatizados y un seguimiento de los productos. Una de las soluciones más importantes que propone Omron es su plataforma para la automatización Sysmac Studio. En el trabajo se ha desarrollado una pequeña guía para entender los criterios más importantes de dicha plataforma.

Por último, se ha cumplido con el objetivo fundamental del proyecto, que es el desarrollo de una aplicación de trazabilidad con el sensor de visión artificial, FQ-M, y una aplicación de Pick&Place, creando un Robot SCARA en 3D con el software Sysmac Studio 3D.

5.2 Trabajos futuros

Con la realización de este Trabajo de Fin de Grado se han planteado varias ideas de mejoras que pueden ser propuestas como líneas de trabajos futuros.

Creación de una aplicación de trazabilidad y Pick&Place real : Aunque en este proyecto se ha utilizado el sensor de visión artificial, FQ-M, de forma real, la aplicación de Pick&Place se ha llevado a cabo de forma simulada creando un Robot SCARA en 3D. Como trabajo futuro se propone el uso del sensor de visión y el robot SCARA de forma real, creando una aplicación de Pick&Place donde ambos dispositivos se comuniquen con la ayuda del controlador de máquina NJ501-4500. Con el software Sysmac Studio se podrán transferir los dispositivos para llevar a cabo la detección de piezas a lo largo de la cinta transportadora. El controlador podrá decirle al robot SCARA cuales son las piezas defectuosas que se deben retirar.

Creación de una aplicación de Pick&Place real con el Robot Delta : En el apartado 2.3 se explicaron las características del Robot Delta y las ventajas que tienen los robots con una estructura de tipo paralelo. Para una aplicación de Pick&Place de colonias el robot ideal es el X-Delta 3+1 ya que permite coger productos de forma más precisa y estable. Como trabajo futuro se propone crear un sistema de Pick&Place real donde el robot Delta recoja una pieza de un punto A y la coloque en un punto B. Para ello se debe utilizar el software Sysmac Studio y establecer la cinemática correspondiente para este robot.

Capítulo 6

Presupuesto

En este capítulo se realiza una estimación de los costes del proyecto, que incluyen tanto los materiales empleados como la mano de obra necesaria para realizar el mismo. El coste del material se divide en dos partes: materiales físicos detallados en la Tabla 6.1 y software representado en la Tabla 6.2.

Tabla 6.1: Coste del material físico.

COSTE DEL MATERIAL FÍSICO				
Dispositivo	Uso (meses)	Duración (años)	Precio	Coste
Portátil ASUS VivoBook	5	4	600,00 €	62,50 €
Robot demo Pick&Place	5	20	Donación a la EPS	- €
Total coste material físico				62,50 €

Tabla 6.2: Coste del software.

COSTE DEL SOFTWARE				
Programa / licencia	Uso (en meses)	Duración (en años)	Precio	Coste
Sysmac Studio	5	4	260,00 €	27,1 €
Sysmac Studio 3D	5	4	252,00€	26,25 €
Total coste software				53,35 €

El coste total del material es la suma de los costes de los materiales físicos y el software, como puede se observa en la Tabla 6.3.

Tabla 6.3: Coste total del material.

TOTAL COSTE MATERIAL	115,85 €
Total coste material físico	62,50 €
Total coste software	53,35 €

Una vez estimados los costes de material físico, se calculan los costes de mano de obra, que engloba tanto las horas de redacción del proyecto como el desarrollo de las aplicaciones. Tabla 6.4.

Tabla 6.4: Coste de mano de obra.

TOTAL COSTE DE MANO DE OBRA			
Descripción	Coste/hora (€)	Horas	Coste
Trabajo de ingeniería	30,00 €	450	13.500,00 €
Redacción del proyecto	20,00 €	200	4.000,00 €
Total coste mano de obra			17.500,00 €

El coste total del proyecto es la suma de los costes de material físico y mano de obra, como se detalla en la Tabla 6.5.

Tabla 6.5: Coste total del proyecto

Total coste realización del proyecto	17.615,8€
Total coste material	115,85 €
Total coste mano de obra	17.500,00 €

En la Tabla 6.6 se muestra el presupuesto de ejecución. Al coste total del proyecto se le añade el beneficio de ejecución del mismo (en este caso del 15 %).

Tabla 6.6: Presupuesto de ejecución.

Presupuesto de ejecución	20.258,2 €
Total coste realización del proyecto	17.615,8 €
Beneficio del proyecto (15 %)	2.624,37 €

Por último, para obtener el presupuesto final del proyecto se debe añadir al presupuesto de ejecución los honorarios de realización del proyecto (7 %) y calcular el IVA del 21 %.

Tabla 6.7: Presupuesto final del proyecto.

Presupuesto final del proyecto	26.228,4 €
Presupuesto de ejecución	20.258,2 €
Honorarios profesionales (7 %)	1.418,1 €
IVA (21 %)	4.552,1 €

Capítulo 7

Pliego de condiciones

El objetivo principal de este proyecto es la realización de dos aplicaciones, aplicación de trazabilidad y aplicación de Pick&Place, que puedan servir para una industria de cosmética. En el proyecto se explican los conceptos de *automatización* y *trazabilidad* para entender la importancia de utilizar las soluciones que ofrecen en todas las industrias.

Para llevar a cabo un sistema de *trazabilidad* en una línea de producción se debe pensar primero en el objetivo principal que se quiere alcanzar con ella. Dependiendo de lo que se quiera conseguir se pueden utilizar diferentes soluciones como tecnología RFID, códigos de barras o sensores de visión artificial. En la aplicación de trazabilidad de este proyecto se utiliza un sensor de visión, de la empresa Omron, que puede ofrecer a las industrias de cosmética los siguiente beneficios [27].

- Analizar la posición de un objeto en particular, coloración, etc.
- Detectar la presencia o ausencia de componentes y piezas.
- Se pueden comunicar con robots autómatas y envían alertas que los ayudan a colaborar en un sistema de Pick&Place.
- Detectar productos dañados, manchados, decolorados, mal posicionados, etc.
- Eliminar la necesidad de inspección visual humana.

De nada sirve un sistema de trazabilidad si no lo sigue después un robot autómatas al que le lleguen las posiciones de las piezas en la cinta que están defectuosas y se deben retirar. La *automatización* de las líneas de producción se hace indispensable para que un sistema de *trazabilidad* sea efectivo y, sobre todo, para lograr la mayor productividad posible dentro de la industria.

Requerimientos a nivel hardware

Como elementos hardware se han utilizado para este proyectos los descritos a continuación:

- **Aplicación Pick&Place de Omron:** La empresa Omron ha donado a la EPS un Robot demo para aplicaciones de Pick&Place. En este proyecto se ha utilizado para poder conectarse al sensor de visión que está integrado en la máquina a través de EtherCAT, para más información consultar el apartado 2.3.
- **Sensor de visión:** Para hacer la aplicación de trazabilidad se ha utilizado el sensor de visión de la empresa Omron FQ-M, figura 2.13, ver apartado 2.3.

Requerimientos a nivel software

Se requiere el uso de diversas herramientas software para la puesta en marcha de este proyecto:

- **Sysmac Studio:** Una plataforma de software, de la empresa Omron, utilizada para la realización de las aplicaciones del proyecto.
- **Sysmac Studio 3D:** Paquete para la plataforma Sysmac Studio que se ha utilizado para la creación del robot SCARA en 3D para la aplicación de Pick&Place del proyecto.
- **Overleaf - LaTeX:** software utilizado para la elaboración de la memoria del proyecto.

Anexos

A BLOQUE DE FUNCIÓN: MC_SetKinTransform

B BLOQUE DE FUNCIÓN: MC_MoveTimeAbsolute

Instruction	Name	FB/ FUN	Graphic expression	ST expression
MC_SetKinTransform	Set Kinematics Transformation	FB		<pre>MC_SetKinTransform_instance (AxesGroup :=parameter, Execute :=parameter, KinTransform :=parameter, Workspace :=parameter, EnableWorkspace :=parameter, MaxVelocityTCP :=parameter, MaxAccelerationTCP :=parameter, MaxDecelerationTCP :=parameter, Done =>parameter, Busy =>parameter, CommandAborted =>parameter, Error =>parameter, ErrorID =>parameter);</pre>

Input Variables

Name	Meaning	Data type	Valid range	Default	Description
Execute	Execute	BOOL	TRUE or FALSE	FALSE	The instruction is executed when <i>Execute</i> changes to TRUE.
KinTransform	Kinematics Transformation	_sMC_KIN_REF	---	---	Set the parameters including the kinematics type and robot link length.* ¹
Workspace	Workspace Parameters	_sMC_WORKSPACE_REF	---	---	Set the range of the workspace.* ²
EnableWorkspace	Enable Workspace	BOOL	TRUE or FALSE	TRUE	Select whether to enable (TRUE) or to disable (FALSE) the workspace check.
MaxVelocityTCP	Maximum Interpolation Velocity	LREAL	Non-negative number	0 * ³	Set a velocity for which maximum interpolation velocity errors of TCP* ⁴ are detected. (unit: mm/s)
MaxAccelerationTCP	Maximum Interpolation Acceleration	LREAL	Non-negative number	0 * ⁵	Set an acceleration rate for which maximum interpolation acceleration errors of TCP* ⁶ are detected. (unit: mm/s ²)
MaxDecelerationTCP	Maximum Interpolation Deceleration	LREAL	Non-negative number	0 * ⁷	Set a deceleration rate for which maximum interpolation deceleration errors of TCP* ⁸ are detected. (unit: mm/s ²)

● _sMC_WORKSPACE_REF

Name	Meaning	Data type	Valid range	Description
WorkspaceType	Workspace Type Selection	_eMC_WORKSPACE_TYPE	100: _mcDelta3 Workspace 101: _mcDelta2 Workspace 102: _mcCartesian3Dworkspace * ² 103: _mcCartesian2Dworkspace * ² 104: _mcHBotWorkspace * ² 105: _mcScaraWorkspace * ³ 500: _mcExpansion1Workspace * ³	Select the workspace type. 100: Delta3 or Delta3R 101: Delta2 102: Cartesian 3D 103: Cartesian 2D 104: H-Bot 105: SCARA robot volume 500: Expansion1 robot volume
WorkspaceParam	Workspace Parameters	ARRAY[0..7] OF LREAL	---	Set the range of the workspace.* ¹

Figura A.1: Manual de usuario NJ Robotics CPU Unit → Details of Robot Instructions (páginas 126-129)

● **_sMC_KIN_REF**

Name	Meaning	Data type	Valid range	Description
KinType	Kinematics Type Selection	_eMC_KIN_TYPE	100: _mcDelta3Type1 101: _mcDelta3Type2 102: _mcDelta2Type1 104: _mcCartesian3DType1 ^{*2} 105: _mcCartesian3DType2 ^{*2} 106: _mcCartesian2DType1 ^{*2} 107: _mcCartesian2DType2 ^{*2} 108: _mcHBotType1 ^{*2} 109: _mcScaraRRPType1 ^{*3} 110: _mcScaraRRPType2 ^{*3} 111: _mcScaraPRRType1 ^{*3} 112: _mcScaraPRRType2 ^{*3} 500: _mcExpansion1 ^{*3}	Select the kinematics type. 100: Delta3 101: Delta3R 102: Delta2 104: Cartesian 3D Robot 105: Cartesian 3D Gantry Robot 106: Cartesian 2D Robot 107: Cartesian 2D Gantry Robot 108: H-Bot Robot 109: SCARA RRP Robot 110: SCARA RRP+R Robot 111: SCARA PRR Robot 112: SCARA PRR+R Robot 500: Expansion1 Robot
KinParam	Kinematics Parameter	ARRAY[0..7] OF LREAL	---	Set the robot arm length and other parameters. ^{*1}

Output Variables

Name	Meaning	Data type	Valid range	Description
Done	Done	BOOL	TRUE or FALSE	TRUE when the settings are completed.
Busy	Executing	BOOL	TRUE or FALSE	TRUE when the instruction is acknowledged.
CommandAborted	Command Aborted	BOOL	TRUE or FALSE	TRUE when the instruction is aborted.
Error	Error	BOOL	TRUE or FALSE	TRUE while there is an error.
ErrorID	Error Code	WORD	^{*1}	Contains the error code when an error occurs. A value of 16#0000 indicates normal execution.

^{*1}. Refer to Section 6 Troubleshooting for details.

● Output Variable Update Timing

Name	Timing for changing to TRUE	Timing for changing to FALSE
Done	When the settings are completed successfully.	<ul style="list-style-type: none"> When <i>Execute</i> is TRUE and changes to FALSE. After one period when <i>Execute</i> is FALSE.
Busy	When <i>Execute</i> changes to TRUE.	<ul style="list-style-type: none"> When <i>Done</i> changes to TRUE. When <i>Error</i> changes to TRUE.
CommandAborted	<ul style="list-style-type: none"> When this instruction is canceled due to an error. When this instruction is executed while there is an error. 	<ul style="list-style-type: none"> When <i>Execute</i> is TRUE and changes to FALSE. After one period when <i>Execute</i> is FALSE.
Error	When there is an error in the execution conditions or input parameters for the instruction.	When the error is cleared.

In-Out Variables

Name	Meaning	Data type	Valid range	Description
AxesGroup	Axes Group	_sGROUP_REF	---	Specify the axes group.

Figura A.2: Manual de usuario NJ Robotics CPU Unit → Details of Robot Instructions (páginas 126-129)

Instruction	Name	FB/ FUN	Graphic expression	ST expression
MC_MoveTimeAbsolute	Time-specified Absolute Positioning	FB		<pre> MC_MoveTimeAbsolute_instance (AxesGroup :=parameter, Execute :=parameter, Position :=parameter, Direction :=parameter, TrajData :=parameter, CoordSystem :=parameter, CSID :=parameter, ToolID :=parameter, BufferMode :=parameter, TransitionMode :=parameter, TransitionParameter :=parameter, Done =>parameter, Busy =>parameter, Active =>parameter, CommandAborted =>parameter, Error =>parameter, ErrorID =>parameter); </pre>

Name	Meaning	Data type	Valid range	Default	Description
Execute	Execute	BOOL	TRUE or FALSE	FALSE	The instruction is executed when <i>Execute</i> changes to TRUE.
Position	Target Position	ARRAY[0..5] OF LREAL	Positive number, negative number or 0	0	Set the target position.
Direction	Direction	ARRAY[0..3] OF _eMC_DIRECTION	0: _mcPositive Direction 2: _mcNegative Direction 4: _mcNoDirection	4	Set the travel direction. 0: Positive direction 2: Negative direction 4: No direction specified
TrajData	Trajectory Data	_sMC_MOVE_TRAJ_REF	---	---	Set the parameters for creating the command value used to move the robot to a specified target position.
CoordSystem	Coordinate System	_eMC_RBT_COORD_SYSTEM	1: _mcRBT_MCS 3: _mcRBT_UCS	1	Select the coordinate system. 1: Machine coordinate system (MCS) 3: User coordinate system (UCS)
CSID	Coordinate System ID	UINT	0 to 15	0	Select the user coordinate system number if you select the user coordinate system (UCS).
ToolID	Tool ID	UINT	0 to 16	0	Select the ID of the robot tool.
BufferMode ^{*1}	Buffer Mode Selection	_eMC_BUFFER_MODE	0: _mcAborting 1: _mcBuffered 2: _mcBlending Low 3: _mcBlending Previous	0	Select the operation that is performed when more than one motion instruction are executed. 0: Aborting 1: Buffered 2: Blending low 3: Blending previous
Transition Mode ^{*2}	Transition Mode	_eMC_TRANSITION_MODE	0: _mcTMNone 11: _mcTMStart Height 12: _mcTMStart RemainingTime	0	Select the trajectory of motion. 0: Transition disabled 11: Start height 12: Start remaining time
Transition Parameter	Transition Parameters	ARRAY[0..7] OF LREAL	Positive number, negative number or 0	0	Set the transition parameters. ^{*3}

Figura B.1: Manual de usuario NJ Robotics CPU Unit → Details of Robot Instructions (páginas 151-153)

● **_sMC_MOVE_TRAJ_REF**

Name	Meaning	Data type	Valid range	Description
MoveTraj Type	Motion Trajectory Type Selection	_eMC_MOVE_TRAJ_TYPE	0: _mcPolynomial3 1: _mcRBT_ModifiedSine	Select the trajectory type used to create the command position for each axis. 0: Polynomial 3 Curve 1: Modified Sine Curve
TrajTime	Trajectory Target Time	UINT	Positive number	Set the time for creating the command position used to move the robot from the current position to the target position. (unit: ms)
MaxVelocity	Velocity Error Detection Value	LREAL	Non-negative number	Set a velocity for which velocity errors are detected*1 during command position creation. (unit: mm/s)
Max Acceleration	Acceleration Error Detection Value	LREAL	Non-negative number	Set an acceleration rate for which acceleration errors are detected*2 during command position creation. (unit: mm/s ²)

Figura B.2: Manual de usuario NJ Robotics CPU Unit → Details of Robot Instructions (páginas 151-153)

Bibliografía

- [1] ROSA FERNANDEZ. Número de empresas activas en españa de 2007 a 2020. *Statista - Global N°1 Bussines Data Platform*, 2021.
- [2] KIMBERLEY MLITZ. Spending on digital transformation technologies and services worldwide from 2017 to 2024. *Statista - Global N°1 Bussines Data Platform*, 2021.
- [3] VANESSA LVEAR-PUERTAS and PAUL ROSERO-MONTALVO. Internet de las cosas y visión artificial, funcionamiento y aplicaciones: Revisión de literatura. *Enfoque UTE*, 8:244–256, 2017.
- [4] JUDITH M MYERSON. *RFID in the supply chain: a guide to selection and implementation*. 2006.
- [5] CARLOS ALEXANDRES FERNÁNDEZ, CARLOS ROGRÍGUEZ-MORCILLO GARCÍA and JOSÉ DANIEL MUÑOZ FRÍAS. Rfid: La tecnología de identificación por radiofrecuencia. 2006.
- [6] OMRON. Automatización. <https://automation.omron.com/es/mx/>, 2021.
- [7] SANTIAGO BLANCO A. JUAN A. ALONSO. Tecnologías de la información y de la comunicación. capítulo 6, programación y control de procesos. *Editorial Ra-Ma*, 2004.
- [8] SINERGES. Robot scara., 2021.
- [9] J. ACEMOGLU, D. ROBINSON. Las instituciones también han sido un ingrediente importante en la consecución del crecimiento económico. «*Why Nations Fail: The Origins of Power, Prosperity, and Poverty, 2*», 2012.
- [10] DIEGO ISTÚRIZ. Expertos advierten sobre la transformación digital: "las empresas que no se adaptan al cambio asumen un riesgo muy alto". «*La Tercera*», 2019.
- [11] PALOMA DURÁN. La importancia de automatizar procesos en una empresa. «*Ricopia Technologies*», 2018.
- [12] FLORENCIA NUÑEZ. ¿qué es la trazabilidad de un producto? «*Bionix*», 2021.
- [13] MANOLO MUÑOZ. Trazabilidad, la trayectoria de un producto., 2014.
- [14] OQOTECH. Iso 22716 - buenas prácticas de fabricación de cosméticos., 2019.
- [15] GUILLERMO RODRÍGUEZ. La trazabilidad de productos cosméticos es ahora elegante y mucho más efectiva con kaiosid., 2020.

- [16] CARLOS RUEDAS. Automatización industrial: Áreas de aplicación en la ingeniería. *Boletín Electrónico*, 10, 2020.
- [17] RAÚL BLANCO DIAZ, JORDI FONTODRONA FRANCOLI, and CARMEN POVEDA MARTÍNEZ. La industria 4.0: El estado de la cuestión. *Economía industrial*, (406):151–164, 2017.
- [18] CARLOS FERNANDEZ BENIQUE. Ventajas del modelado 3d en proyectos de ingeniería. «*CACCA*», 2019.
- [19] MARTA BARAMBIO MARTINEZ. La importancia de la automatización desde tiempos remotos. «*Intertronic*», 2018.
- [20] JOAQUIN EUGENIO DE LIMA. ¿cuáles son los beneficios de la trazabilidad para su empresa? «*my ABCM*», 2021.
- [21] CESAR IVÁN SOSA LEONARDO. Propuesta de un sistema de trazabilidad de productos para la cadena de suministro agroalimentaria. *Universitat Politècnica de Valencia*, 2017.
- [22] JOSÉ MANUEL HUIDOBRO. Código qr. *Bit, dic.-ene*, 172:47–49, 2009.
- [23] JUSTIN BROWN. Códigos de barras industriales. *COGNEX*, 2020.
- [24] JORGE ALBERTO ALVARADO SÁNCHEZ and other. Sistema de control de acceso con rfid. *México DF*, 2008.
- [25] RODRIGO FERNÁNDEZ FERNÁNDEZ. Optimización de la trayectoria de un robot scara paralelo. *Universidad Politècnica de Valencia*, 2020.
- [26] DALIMIR ORFANUS, REIDAR INDERGAARD, GUNNAR PRYTZ, and TORMOD WIEN. Ethercat-based platform for distributed control in high-performance industrial applications. In *2013 IEEE 18th Conference on Emerging Technologies & Factory Automation (ETFA)*, pages 1–8. IEEE, 2013.
- [27] VINSSA. Ventajas de los sistemas de visión artificial., 2021.

Universidad de Alcalá

Escuela Politécnica Superior



ESCUELA POLITECNICA
SUPERIOR



Universidad
de Alcalá